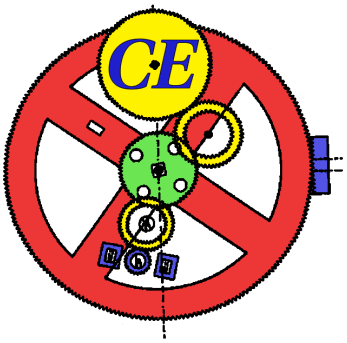


Computer Engineering
Mekelweg 4,
2628 CD Delft
The Netherlands
http://ce.et.tudelft.nl/

2010

MSc THESIS

Evaluation Methodology and Systematic Selection of Microcontrollers for Delfi-n3Xt Nanosatellite

Armin Noroozi

CE-MS-2010-07

Abstract

The success of Delfi-C^3 made it possible to define a complete roadmap of nanosatellite missions. As with Delfi-C^3 multidisciplinary teams of students from Aerospace Engineering and Electrical Engineering Faculties will be involved. The first satellite is Delfi-n3Xt (pronounced as *delfi next*).
In the design of the Delfi-n3Xt nanosatellite, power consumption and performance are the major criteria. One of the important investigations that has to be done is to find the most suitable microcontroller for this satellite in order for it to successfully accomplish its mission. Based on the mission specifications of the Delfi-n3Xt nanosatellite and its requirements, a study will be done on the commercially available microcontrollers. A selection will be made between them and based on the result, the best candidate will be chosen. Therefore an evaluation methodology is proposed. This methodology is based on the requirements of this satellite. For each requirement, a criterion will be defined and a weight (depending on on its importance) will be associated with it. Every microcontroller receives points, based on how well it fulfills a compared criterion. By the end, a ranking system will be formed to order the microcontrollers. The goal of this system would be to assign scores to each MCU based on their points, to find the most suitable microcontroller for this project.
The microcontrollers will be selected and their general characteristics and specifications will be discussed. Following the steps provided in the evaluation methodology, the microcontrollers will be compared in each criterion, with a discussion on their differences. They will get evaluated and will receive points. By the end, the ranking system will be applied and the microcontroller with the highest score will be chosen as the final microcontroller.
After the evaluation, a measurement board using the same microcontroller will be designed, built and tested. The test results will be briefly described. This board would be widely useful for creating and testing different flight software modules for the satellite. More precisely, it includes communication within the satellite (between modules), satellite with the ground station, reading payload data within the satellite and other necessary programs.

TU Delft

Faculty of Electrical Engineering, Mathematics and Computer Science

Evaluation Methodology and Systematic Selection of Microcontrollers for Delfi-n3Xt Nanosatellite

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

by

Armin Noroozi
born in Tehran, Iran

Computer Engineering
Department of Electrical Engineering
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Evaluation Methodology and Systematic Selection of Microcontrollers for Delfi-n3Xt Nanosatellite

by Armin Noroozi

Abstract

The success of Delfi-C^3 made it possible to define a complete roadmap of nanosatellite missions. As with Delfi-C^3 multidisciplinary teams of students from Aerospace Engineering and Electrical Engineering Faculties will be involved. The first satellite is Delfi-n3Xt (pronounced as *delfi next*).

In the design of the Delfi-n3Xt nanosatellite, power consumption and performance are the major criteria. One of the important investigations that has to be done is to find the most suitable microcontroller for this satellite in order for it to successfully accomplish its mission.

Based on the mission specifications of the Delfi-n3Xt nanosatellite and its requirements, a study will be done on the commercially available microcontrollers. A selection will be made between them and based on the result, the best candidate will be chosen. Therefore an evaluation methodology is proposed. This methodology is based on the requirements of this satellite. For each requirement, a criterion will be defined and a weight (depending on on its importance) will be associated with it. Every microcontroller receives points, based on how well it fulfills a compared criterion. By the end, a ranking system will be formed to order the microcontrollers. The goal of this system would be to assign scores to each MCU based on their points, to find the most suitable microcontroller for this project.

The microcontrollers will be selected and their general characteristics and specifications will be discussed. Following the steps provided in the evaluation methodology, the microcontrollers will be compared in each criterion, with a discussion on their differences. They will get evaluated and will receive points. By the end, the ranking system will be applied and the microcontroller with the highest score will be chosen as the final microcontroller.

After the evaluation, a measurement board using the same microcontroller will be designed, built and tested. The test results will be briefly described. This board would be widely useful for creating and testing different flight software modules for the satellite. More precisely, it includes communication within the satellite (between modules), satellite with the ground station, reading payload data within the satellite and other necessary programs.

Laboratory : Computer Engineering
Codenumber : CE-MS-2010-07

Committee Members :

Advisor:	Georgi Gaydadjiev, CE, TU Delft
Chairperson:	Koen Bertels, CE, TU Delft
Member:	Arjan van Genderen, CE, TU Delft
Member:	Chris J.M. Verhoeven, EE, TU Delft

To my family for their endless love and support

Contents

List of Figures

List of Tables

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Georgi Gaydadjiev for his immense help, guidance, stimulating suggestions and encouragement with this thesis work. It was a great pleasure to do thesis under his supervision.

I am grateful for my family, for their endless love and support during my studies here in TU Delft. Specially my brother Arash, who supported and guided me throughout my studies and the writing of my thesis report.

Armin Noroozi
Delft, The Netherlands
May 11, 2010

1 Introduction

Throughout history, man has gained information regarding the formation of stars, planets and the rules governing them, without leaving the Earth surface. It began by using telescopes to discover objects positioned far away from Earth in outer space. This led to creation of observatories. During the World War II, missiles were created that could fly over long distances. Sending long range rocket-missiles, led to launching rockets to space after the World War II [56]. After this event, attempts were made in order to successfully send objects into space that will orbit the Earth. Hence, satellite development started.

The first satellite successfully launched to space was the 83.6 kilograms spherical shaped (53 centimeters in diameter) satellite, made by USSR, named *Sputnik 1.* It was launched on 4th of October 1957 and orbited the Earth on the altitude of 900 kilometers above the Earth surface (Exosphere) at the speed of 8000 m/s [30]. This was the beginning of the *Space Race* [54]. Sputnik 1 decayed on 4th of January 1958 and burnt up in the atmosphere [54][19]. After the launch of Sputnik 1, US successfully launched the *Explorer 1* satellite on 31st of January 1958.

Sputnik 1 was equipped with a battery and two radio transmitters emitting telegraph formed signals continuously. Its only task was to send these radio signals from its orbit to Earth. These signals could be received by a broad range of radio amateurs [30]. After the battery died on 26th of October 1957, the signal transmission stopped [54][19]. Afterwards USSR launched the *Sputnik 2* (3rd of November 1957) which carried a dog (named *Laika*) into space, monitoring her vital signs and transmitting them back to Earth via telemetry signals and *Sputnik 3* (15th May 1958) in order to study the Ionosphere [19].

Throughout all these years, new technologies have been developed in different fields of science (such as microelectronics, astrodynamics, celestial mechanics, optical physics, orbital mechanics, etc.). These technologies have provided us the ability to achieve same tasks and missions that required lots of resources to accomplish, in a more simple way with better results than before. New satellites have been developed for different purposes such as telecommunication, Earth and space observatory, weather prediction system, etc.

The new technologies in propulsion systems and material science, together with the new and improved energy sources, allow the rockets to fly over a longer distance. With the help of space qualified and more reliable computer systems and sensors, the satellites are able to calculate their position and trajectories better than before and remain in the orbit for a longer period. By using the growing technology of solar cells, low power microelectronic designs and high performance microcontrollers, more electrical power can be provided for satellites, allowing them to accomplish more tasks in a shorter period.

In the past, only military or very big agencies could make or fly satellites. Nowadays, students at universities can do the same as a part of their studies. These universities provide the required facilities for the students to manufacture and do research on new

methods to create various spacecrafts, that use the new technologies.

Satellites are divided into several categories which indicate their primary mission. A few examples of these categories are weather satellites, space stations, navigational satellites, biosatellites, miniaturized satellites, astronomical, reconnaissance, Earth observation satellites, etc. [53]

In satellite classifications, a general method has been adopted recently. This method, classifies the satellites in terms of their mass. In this classification, wet mass[1] of more than 1000 kg is considered as a *Large satellite*. Wet mass between 500 kg and 1000 kg defines a *Medium sized satellite*. Mass of less than 500 kg is called a *Miniaturized satellite*. [36]

Miniaturized satellites are the low-weight small sized satellites which weigh at most 500 kilograms. Although all these satellites can be referred to as small satellites, further classifications have been made in order to categorize them based on their mass [50]. These categories are as follows:

- Minisatellite: such satellites have a wet mass of 100 kg to 500 kg. They are usually called *small satellites*. In comparison to the larger satellites, these satellites are usually more simple while using the same technology as the large satellites [50].
- Microsatellite: satellites made with a mass between 10 kg to 100 kg fall under this category. These satellites are also known as *microsat* and sometimes the generic term *small satellite* is used to represent them. [50].
- Nanosatellite: these satellites (aka *nanosat*) have a wet mass of 1 kg up to 10 kg [50].
- Picosatellite: satellites with mass between 0.1 kg up to 1 kg [50].
- Femtosatellite: Mass less than 100 grams.

There is another type of satellite classification which applies to a satellite with maximum weight of 1 kg and an exact volume of one liter ($10\times10\times10$ cm^3 cube) that uses commercial off-the-shelf components inside. This type is called a *CubeSat*.

As it has been mentioned before, the type of a satellite can be defined based on its primary mission, size or mass. In the following sections, Delfi-C^3 and Delfi-n3Xt nanosatellites are introduced. These sections cover general information regarding these two nanosatellites, such as their classification, mission and payloads.

1.1 The Delfi-C^3 nanosatellite

The chair of System Integration was approached by Dutch Space in November 2004, to discuss the possibility of testing a new type of thin film solar cells in the space environment [15]. As a follow-up, the same week contact was sought with TNO's[2] Science and Industry (S&I) Department and the Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS). Both reacted positively to corporate in the proposed

[1] Wet mass is the mass of a satellite including its fuel.
[2] TNO is Netherlands Organization for Applied Scientific Research.

mission. EEMCS is interested to provide an advanced transceiver as payload, while TNO-S&I is interested in a flight opportunity for an analog sun sensor that operates wireless and autonomous. [15]

The basic design principle of this satellite was the CubeSat concept. With this design in mind, the Delfi-C^3 project was created. This project started in November 2004 and Delfi-C^3 was successfully lunched on 28 April 2008 as tertiary payload on-board of an Indian PSLV[3] launcher. [15] Delfi-C^3 still remains fully functional after two years in space.

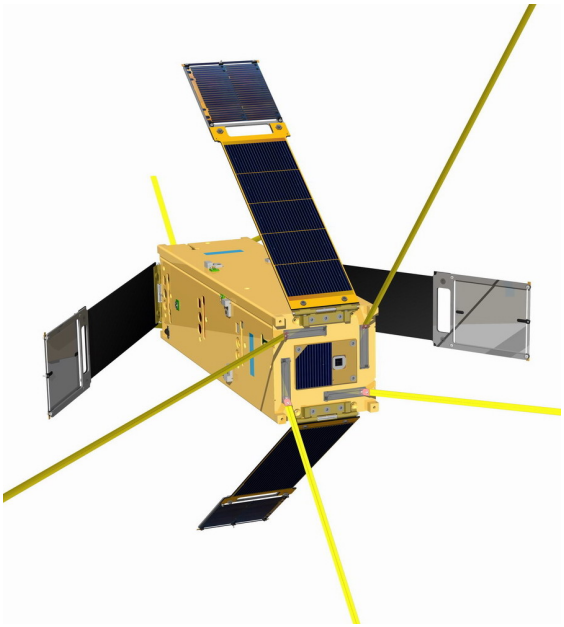


Figure 1.1: Delfi-C^3 nanosatellite

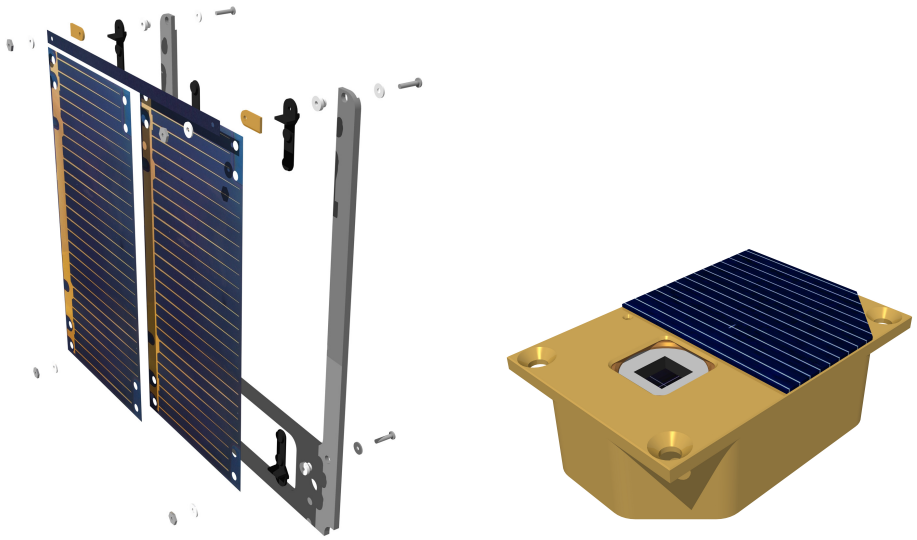
1.1.1 Mission and Payloads

This satellite (Figure 1.1) was created as a platform to test new devices in-orbit. The mission duration of this satellite was defined six months. The following list of payloads and experiments are present on board of Delfi-C^3:

- Thin Film Solar Cells (TFSC) (by Dutch Space)
- Autonomous Wireless Sun Sensor (AWSS) (by TNO)
- Advanced Transceiver Experiment
- In-orbit test of a Radio-Amateur Platform (RAP)

In total there are four pairs of two TFSC attached to this satellite. The TFSC is 80×40 mm, made on a titanium foil with 50 μm thickness. Figure 1.2(a) shows the

[3]Polar Satellite Launch Vehicle, developed by Indian Space Research Organization. [33]



(a) Thin Film Solar Cells in suspension frame [46]

(b) Autonomous Wireless Sun Sensor [46]

Figure 1.2: Delfi-C^3 payloads

rendered model of this solar cell. Figure 1.1, shows these solar cells attached to the satellite.

A combination of two Autonomous Wireless Sun Sensors and one RF-receiver, is on this satellite. It weighs 75 grams and has a volume of about 60×40.5×17.8 mm. Figure 1.2(b) shows a 3D rendered model of this sensor. These two sensors are positioned at the two ends of this satellite. One of these sensors is visible in Figure 1.1.

1.2 The Delfi-n3Xt nanosatellite

After the meeting that was held on October 2007 by the chair and staff of Space Systems Engineering (SSE) and staff of Electrical Engineering department involved in the development of Delfi-C^3, at Aerospace Engineering faculty, continuation with a development line on nanosatellites was concluded focusing on the miniaturization of satellite technology and the educational aspects of a real satellite mission [16]. The successor of Delfi-C^3 nanosatellite is named Delfi-n3Xt (pronounced as *delfi next*). The '*next*' is to indicate that this satellite is the successor of the previous Delfi nanosatellite and the '*3X*' stands for three-axis stabilized [16]. Figure 1.3 shows a 3D rendered model of this satellite. In this figure, inside of the satellite and four solar panels attached to it (blue color) are visible.

1.2.1 Mission

Delfi-n3Xt mission is best described by the following goals:

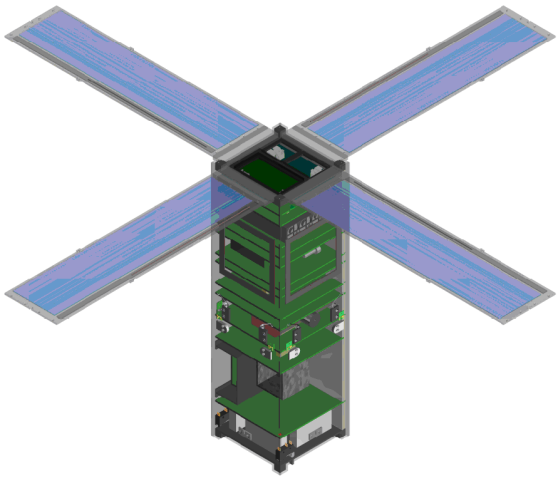


Figure 1.3: Delfi-n3Xt nanosatellite

- Pre-qualification of a mirco-propulsion system from TNO, TU Delft and UTwente
- Pre-qualification of a Multi-functional Particle Spectrometer (MPS) from Cosine Research BV
- Scientific Radiation Experiment of Si-solar cells from DIMES
- Qualification of a high-efficiency communications platform from ISIS BV
- Proof-of-concept for a radiation risk-free implementation of commercial solid-state data storage devices, provided by NLR
- Implementation of 3-axis active attitude control
- Providing high data-rate (> 9.6 kbps) links[4]
- A single-point-failure-free EPS with energy storage

1.2.2 Payloads

Delfi-n3Xt nanosatellite Command and Data Handling Subsystem (CDHS) is designed in such a way that will assure safety of the satellite during the mission. Payloads are designed modular. The modular design means that satellite payloads are designed independently in modules and every module on the satellite is responsible for a certain task. In total there are five payloads. The following is a list of the different modules including payloads that will be on the Delfi-n3Xt nanosatellite [16]:

- Electrical Power Subsystem (EPS)
- On Board Computer (OBC)

[4]S-band transmitter (> 9.6 k) will only be taken on board under strict conditions.

- Attitude Determination and Control Subsystem (ADCS)
- COMMS 1 (Communication Subsystem)
- COMMS 2 (Communication Subsystem)
- Micro Propulsion Flight Experiment ($T^3\mu PS$) (by TNO)
- Multi-functional Particle Spectrometer (MPS) (by Cosine Research BV)
- Solar Cell Degradation Measurement (TFSC) (by DIMES[5], TUDelft)
- VHF-UHF Transceiver (by ISIS BV)
- COTS data-storage experiment (by NLR[6])

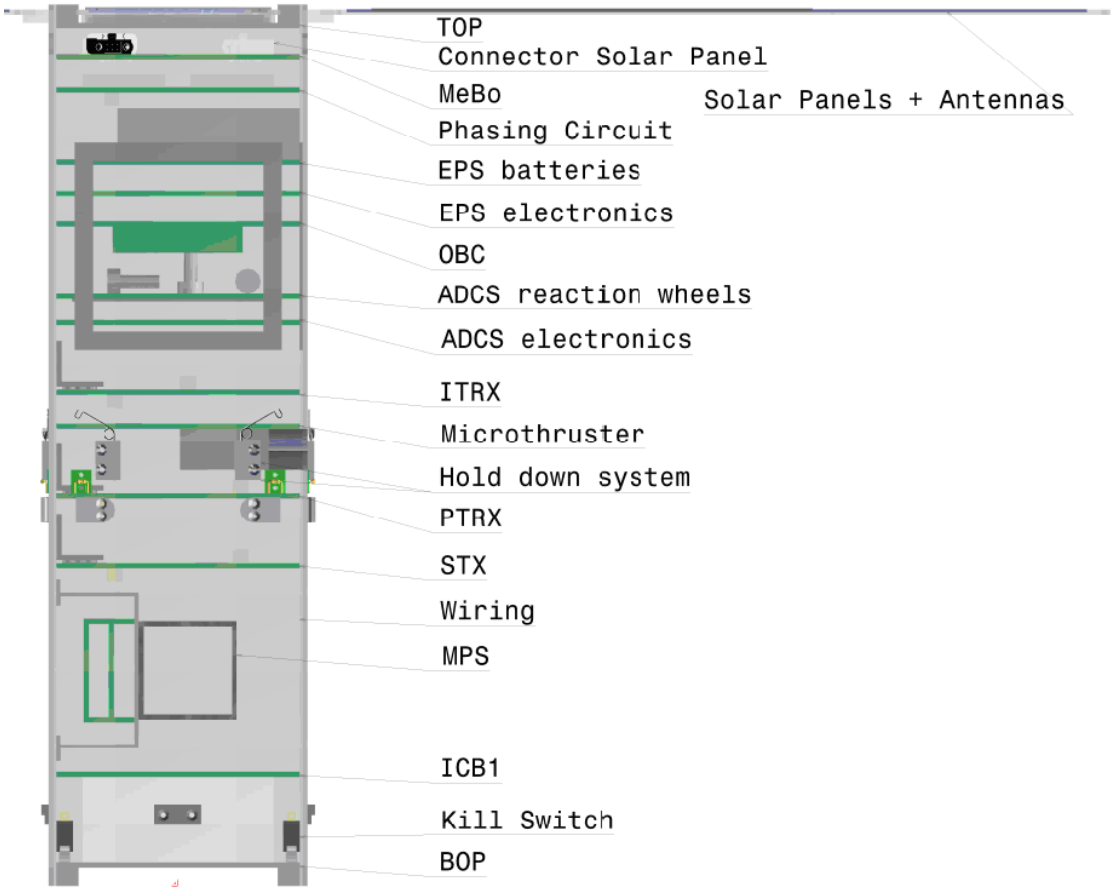


Figure 1.4: Delfi-n3Xt payloads

[5]Delft Institute of Microelectronics and Submicrontechnology
[6]Nationaal Lucht- en Ruimtevaartlaboratorium

Figure 1.4 shows different payloads and the physical structure of the Delfi-n3Xt nanosatellite. EPS provides the power required for all the subsystems and payloads during the mission. OBC is the main computer of this satellite. It acts as the brain, communicates with all subsystems and synchronizes them with each other. ADCS controlls the position and trajectory of this satellite. Communication Subsystem in this satellite is divided into different modules: PTRX (Primary Transceiver), ITRX (ISIS Transceiver) and STX (S-band Transmitter). PTRX is used to handle the communication and command link between the satellite and ground station. ITRX is a VHF-UHF transceiver designed by ISIS. STX is an S-band transmitter on board the satellite which sends collected data to ground station.

1.2.3 Requirements

The following, is a list of the tasks specified for the OBC. The OBC microcontroller must be able to do these tasks without any problems [20].

- System health check and status monitoring
- Data transfer control

  From payloads to OBC

  From OBC to PTRX. Data will be transmitted from PTRX (10 kbit/s)

  From OBC to data storage.

  From data storage to STX.
- Request data and Tele-Commands from systems
- Schedule the activation of systems (delay loops, RTX request, etc.)
- Validate commands
- Manage the data storage (flash memory or RAM or other types of memory)
- Acknowledge commands
- Format data packages (frame, time tag, etc.). This is for data storage and transmission to the ground station.
- EDAC for storage. (Error detection and correction) (t.b.d.).

1.3 Conclusion

As it can be seen from the mission definition of both Delfi-C^3 and Delfi-n3Xt nanosatellites, there are many differences in their design, mission and requirements. Delfi-n3Xt is going to provide more features while keeping the physical size of Delfi-C^3 nanosatellite.

1.4 Thesis Framework

This section describes the framework of this thesis. The goal of this thesis is to find the most suitable microcontroller for the Delfi-n3Xt nanosatellite. To achieve this goal several steps have been taken. These steps create the chapters of this thesis:

Chapter 2 begins with history of processor architectures. It gives a brief description on the developed processor architectures. This study is done to show what researches and developments have been done until now, in processor design.

Chapter 3 will propose an evaluation methodology for selecting a microcontroller for this satellite. In this methodology different analysis criteria will be defined. These criteria are based on the requirements of this satellite. Each criteria will receive a weight to indicate its importance. At the end, a ranking system will be proposed and presented. This system will rank different microcontrollers in each criteria. The result of this system is a table which indicates the most suitable microcontroller for this satellite.

Chapter 4 will be the result of applying the methodology presented in chapter 3, on commercial off-the-shelf microcontrollers. It will begin by selecting several microcontrollers as candidates and providing information on each one of them. Afterwards, trade-offs will be made in all criteria between the microcontrollers. A table will be created for every comparison, which will hold points for each microcontroller. After the comparisons, the ranking system will be applied and the overall score of each microcontroller will be calculated. Based on the result of this system, the most suitable microcontroller will be chosen.

After the microcontroller is chosen, a measurement board is developed. Chapter 5 will cover information on the steps taken to design, create and test this measurement board and the chosen microcontroller along with the test results.

In Chapter 6 conclusion and recommended future work is provided. Conclusions are written as a summary of the thesis work.

Background and Related Research

2

To find the most suitable microcontroller, an understanding of a microcontroller is necessary. This chapter begins with a brief history and background on different processor architectures. The main focus here, is on clarifying a definition and description of each architecture along with their characteristics.

2.1 Processor Architectures History

Computers built in the early 1950s, used based-10 instead of base-2, as their default numbering system. After further developments, binary based arithmetic designs were developed inside the processor architectures. One major drawback[1] of these processors was that the programs that could run on one processor, could not run on the others. After further development of these architectures, a new design was required that could support all forms of processing. This new design contained instruction sets that were designed in order to manipulate simple binary numbers, text, floating points and decimal arithmetic coded in binary. Almost all processors designed afterward included these ideas. This basic set is called a *Complex Instruction Set Computer* (CISC). In this design, an instruction has access to the register or memory in several different ways. This made CISC easier to program, since the programmer had to remember and use almost hundred instructions and a set of addressing modes, instead of thousands of distinct instructions. [49]

CISC became a powerful concept for computer design, because of its small sized code. This allowed a program with many capabilities to be stored in a small sized memory. Since there was a semantic gap[2], compilers were being developed that could create a better code with a more richer instruction set. [49]

According to the researchers from UC Berkeley and IBM in early 1980s, most of the computer language compilers and interpreters that were created, used only a small subset of the instructions of CISC. They concluded that most of the processor power, was ignored. Therefore, the computer designs should be made simpler and less orthogonal [3]. By then, the processor calculations became faster than the memory access speeds. In order to achieve this, a new processor architecture was built, which could cache the intermediate results in the registers, under compiler control. This design was called a *Reduced Instruction Set Computer* (RISC): a very simple core processor which runs at a very high speed. In general, RISC has more registers than CISC with less orthogonal instruction sets, a few load and store instructions in order to read and write data from/to

[1] This is still the case with many processors.

[2] Semantic gap is a defined distance between machine language and a high-level language. [49]

[3] Instruction set of a processor is orthogonal if any instruction is able to access any type of data by using any addressing mode. [52]

memory and it supports the same operations that the compilers were written for and supported. [49]

RISC also became popular and has been used in the design of the microcontrollers ever since. Since its number of instructions is less than the ones defined in CISC, the size of the programs written for them became larger than the size of the same program written for a CISC machine. This issue limited the fast memory handling of a RISC based processor, while handling a larger code size. [49]

A company named Acorn was about to make a new platform for computers. Therefore they tested all available processors and their engineers came across the RISC project. They began to design a new architecture that could have a higher performance than the RISC machines created by the Berkeley students. The main goal in their design was to achieve a low-latency interrupt handling like the MOS Technology 6502 processor[4]. In 1983 the official *Acorn RISC machine* (ARM) project was started. In 1985 ARM1 was produced and the following year ARM2 was available. [47]

CISC and RISC have been the available choices ever since. Many new architectures have been built upon these concepts. The ARM architecture has gone under lot of development and modifications. New processor architectures have been built based on these concepts such as the x86. The x86 architecture has become the core for many developed processors such as Intel 8086, Intel 80186, Intel 80286, Intel 386, Intel 486, Intel Pentium, Intel Core 2, Intel Atom, Intel Core i7, Intel Core i5, AMD Am386, AMD K6/2, AMD Phenom and many more. More information on the processor architectures, processors and microcontrollers, is provided in the following sections.

2.2 Instruction Set Architecture (ISA)

A processor design is divided into three steps: *architecture*, *implementation* and *realization*. Table 2.1 shows an overall difference between these three steps. Architecture step, is concerned about the specifications of the *function* provided by the processor to the programmer. In the implementation stage, designer finds the ways this functionality should be implemented. In this stage, the designer considers all factors of the final design regarding the *cost/performance* ratio from the beginning of the design. Realization step is concerned about the type of components and technologies being used (e.g. CMOS) and their placement on the design. In addition, in terms of absolute time, it deals with the time a signal should pass a component. Drawings, lists and instructions are the outputs of the realization design of the processor production process. [24]

In order for the programmer to interact with the processor hardware, a set of instructions are provided that the processor can carry out. Instruction Set Architecture, specifies these instructions, which clarifies different types of operations the processor can perform along with the data types needed by each operation. Also, addressing modes (mechanisms for locating the data values) are specified by ISA for the processor. [24]

ISA defines the processor personality and how it functions [18]. The idea of ISA is to have a common platform to execute a program. The ISA layer exists to provide the

[4]An 8-bit processor desinged in 1975 for MOS Technology. It was the least expensive full featured microprocessor, comparable with the ones from Motorola and Intel.

Design domains	Questions	Concerns
Architecture Functional appearance (to the systems programmer)	What?	Function
Implementation Logical structure (performs the architecture)	How?	Method
Realization Physical structure (embodies the implementation)	Which? Where? When?	Realization

Table 2.1: Domains of computer design [24]

details needed by a programmer about the machine. A compiler, translates the codes written in a programming language (C, Java, Fortran, etc.) to the equivalent machine language, that can run on ISA-level logic processor [18].

Processor designs can be divided into two categories at the ISA level: CISC and RISC. In different ISAs, different number of addressing modes, operations and data types are specified. For example in CISC, a simple ALU instruction is combined with one or more memory operations. In this way, the processor does both operations by receiving one instruction. This has led to existence of several instructions with different lengths. In RISC, each of the instructions are designed to carry out a specific operation such as "load" or "add" and have a fixed length. [18]

2.3 Microcode

A microcode (also called microprogram) is a very small code that acts as a run-time interpreter for instructions. It takes the complex instructions and generates a sequence of simple instructions [18]. Microcode is written in the design step of a processor. It is usually stored in PLA (Programmable Logic Array) or ROM (Read Only Memory). It resides between the instructions and the underlying electronics and makes it possible to make modifications to the instructions without modifying the electronics design of the processor. It also makes it possible to improve the processor electronics without changing the instructions.

2.4 Processor Architectures

There are many processor architectures available such as the Z80, x86, M68000, MIPS, ARM, etc. In the following subsections, a few of these architectures are discussed.

2.4.1 Processor Basics

A processor is built up of three main parts: a set of registers, an arithmetic logic unit (ALU) and a control unit (CU). Different register sets can exist in processors, which makes processor architectures differ from one another. Register sets are a combination of general-purpose registers and special-purpose registers. General-purpose registers, can be used for any purpose. They can be assigned to various functions of a program. Special-purpose registers are designed to have a specific function. They are restricted to certain functionalities (such as Program Counter, Stack Pointer, etc.). In addition to the difference in usage of registers, processors may differ in the number and size of registers. In an 8-bit design, registers with 8-bit length could be used to store 8-bit data. In a 16-bit design, it is possible to use 16-bit registers, or use two contiguous 8-bit registers to hold 16-bit data. [1]

2.4.2 Von Neumann Architecture vs. Harvard Architecture

The von Neumann architecture consists of a single storage and CPU. The storage structure holds both data and instructions. A single bus connects the memory to CPU. Therefore, the CPU is able to either read an instruction from memory or do data read/write operation on the memory. In contrast to this architecture, is the Harvard architecture. It has separate memories for data and instruction, which allows it to read instructions and access memory to read/write data simultaneously. [26]

The Von Neumann architecture has a benefit over the Harvard architecture. It allows the instructions to be treated as data. In this way, the code (instructions) can be read directly from the disk and executed. Nowadays, the processors that use the Harvard architecture, actually use the *Modified Harvard Architecture*. As the name suggests, this architecture is an improvement to the pure Harvard Architecture. It unifies a large portion of data and instruction address space (just like von Neumann).

To distinguish Modified Harvard architecture from Harvard and von Neumann architectures, the following characteristics can be used [51]:

- **Address space occupied by data and instruction memories**

  Harvard architecture defines separate address 'zero' for data and instruction memories. This address points to the beginning of data memory address space in data memory space. In instruction memory space, this address points to the beginning of the instruction memory address space. In von Neumann and Modified Harvard architectures, data and instruction are stored in a single address space. Therefore, the address 'zero' points to the beginning of the same memory address space.

- **Hardware connection between CPU and data and instruction memories**

  In pure Harvard architecture, separate memory pathway provides a simultaneous access to data and instructions. This speeds up the performance. In the Modified Harvard architecture, there are separate paths for CPU cache and the unified memory space. The von Neumann architecture is without the presence of CPU cache. Therefore, there is only one path between the unified memory space and CPU.

- **Methods to access data and instruction memories**

  Due to limitation in technology, the original Harvard machine, used punched paper tape to store instructions and electro-mechanical counters to store data. Nowadays, a Harvard based machine such as PIC microcontroller, may store instructions in a 12-bit wide Flash memory, while using an 8-bit wide SRAM to store data. In a modified Harvard or von Neumann microcontroller (e.g. ARM7TDMI), a uniform access is provided to Flash and SRAM (8-bit for both).

Now that the two widely used processor architecture designs are clear, the CISC and RISC concepts have to be explained.

2.4.3 Complex Instruction Set Computer (CISC)

The Complex Instruction Set Computer (CISC) is a type of computer Instruction Set Architecture (ISA) that has the ability to execute a series of low level instructions in one single instruction. Depending on the specific instruction, it could take more than one cycle to complete its operation.

Since most of the early machines were programmed in the assembly language by the programmer, the CISC was designed with many instructions to make this task easier for the programmer. Also the low amount of memory (slow memory) lead to the design of CISC in the way it was. Some of the common characteristics of CISC based architectures, would be their 2-operand instruction format, several memory addressing modes and variable length instruction. The length of an instruction varies depending on the addressing mode.

Because of the limited area on the chip and lots of instructions available that can operate directly from the memory, a small number of general purpose registers have been implemented. For example, the Intel 8086 architecture implements fourteen 16-bit registers. The Intel 80386 expanded all the 16-bit registers of 8086 to 32-bit registers, except for the segment register (remained 16-bit). The following subsections provide more information on a few CISC based architectures.

2.4.3.1 Motorola 68000

The 68000 is a 16-bit processor architecture developed by Motorola. Its hardware architecture consists of a 16-bit bidirectional data bus and a 24-bit address bus [4]. There are eight 32-bit data registers (D0-D7), seven 32-bit address registers (A0-A6), one 32-bit program counter, two 32-bit stack pointers and one 16-bit status register available on this processor. Although this processor contains 32-bit data registers, it is considered as a 16-bit processor. This is due to the fact that its data bus is only 16-bit wide. [31]

2.4.3.2 x86

The x86 is the instruction set architecture family based on Intel 8086 processor core. The letter 'x' is used to indicate backward compatibility of the new processors. 80x86, 80x286, 80x386, i486, Pentium, Core 2 Duo, Quad Core and Core i7 are examples of this processor architecture family. The first 8086 processor was designed with an 8-bit data

bus. 286 used 16-bit data bus. 386 and 486 designs use 32-bit data bus. Pentium and above use 64-bit data bus [1].

Each of the 80x86 successors, have added several numbers of architectural features to its predecessor. The 8086 based processors built in 1978, had a performance of approximately 0.33 MIPS[5]. In 1989, the i486 processors reached the performance of approximately 12 MIPS, along with new features, such as multiprocessor support, integrated instruction/data cache and page based memory management. [3]

Since there are spacecrafts built with a 68k processor inside, this processor is chosen as one of the microcontroller selection categories. Later, a microcontroller that uses the latest 68000 processor as its core, is selected.

2.4.4 Reduced Instruction Set Computer (RISC)

The main goal of RISC ISA, was to have less instructions in order to keep the software code as simple as possible. The important characteristics of the RISC which separate it from CISC beside the fewer number of instructions, would be its larger number of registers, execution time of one cycle and the pipelining technique used in the architecture. A certain amount of processor speed is spent on the communication between processor and memory. In order to reduce the communication and interaction between the processor and the memory, large number of registers is included in a RISC-based design. For example in the design of MIPS I, there are 32 general purpose registers. The clock per instruction (CPI) of a RISC processor is one cycle which is achieved by optimizing each instruction on the processor. Unlike CISC, a RISC-based processor architecture contains no microcode stage.

RISC characteristics can be summarized as follows:

- Large uniform register farm
- Load/Store architecture which makes data-processing operations to work only on registers and not directly on the memory contents
- Fixed length and uniform instruction fields which simplifies instruction decoding
- Simple addressing modes, with all load/store addresses being determined from the instruction fields and register contents.

2.4.4.1 PowerPC (Performance Optimization With Enhanced RISC-Performance Computing)

PowerPC processor contains three independent execution units: branch processing unit, fixed-point unit (a.k.a instruction unit) and floating-point unit. The branch processing unit processes the branch instructions. The fixed-point unit, executes fixed point and load/store instructions. The floating-point unit handles the process of floating-point instructions. [17]

[5]Million Instructions Per Second

2.4.4.2 SPARC (Scalable Processor Architecture)

SPARC is a RISC based instruction set architecture with 32-bit integer and 32-, 64- and 128-bit floating point data types. It defines 69 32-bit wide basic instruction operations. This type of processor consists of an integer unit, a floating point unit and a co-processor. The co-processor is an optional unit within this architecture. Each of these units have their own register sets. All registers are 32-bit wide (except for co-processor). [34]

Overall operation of the processor is controlled by the integer unit. It controls floating point unit and co-processor instruction execution while maintaining program counters computing memory addresses for load and store. This unit contains 32-bit general purpose r registers (number of registers may vary between 40 and 520). [34]

The floating point unit consists of 32 32-bit floating point f registers. These registers can hold up to 32 single-precision, 16 double precision and 8 quad-precision values. Floating point load/store instructions are used for data movements between this unit and the memory (address calculated by the integer unit). [34]

The instructions are categorized into 6 classes: Load/Store, Arithmetic/logical/shift, Control transfer, Read/Write control register, Floating-point operate and Co-processor operate. [34]

2.4.4.3 MIPS (Microprocessor without Interlocked Pipeline Stages)

A MIPS based CPU consists of 32 32-bit general-purpose registers, two 32-bit registers to hold multiplication and division operation results and one 32-bit program counter register. The processor has 32-bit instructions. They can be categorized into the following classes: Load/Store, Computational, Jump and Branch, Co-processor and Special instructions. [28]

2.4.4.4 Advanced RISC Machine (ARM)

The ARM architecture is based on Reduced Instruction Set Computer (RISC) processor architecture principles and has the RISC architecture features implemented in it. In addition to the RISC features, ARM has the following features in its architecture:

- Provides control over the Arithmetic Logic Unit (ALU) and the shifter
- Load and store multiple instructions in order to maximize the data throughput
- Conditional execution of most of the instructions in order to maximize execution throughput
- Auto increment/decrement of the addressing modes for loop optimizations.

Control over both ALU and the shifter has been provided for most of the data processing instructions in order to maximize the use of the ALU and shifter.

One of the features of the ARM is its pipelining. pipelining has been implemented in its design in order to provide a continues operation for all processing parts and memory systems. Features of the ARM are discussed in the following subsection.

2.4.4.5 ARM features

ARM is a 32-bit RISC processor with improved architecture than other RISC-based processors. In this section, important aspects of ARM are discussed.

ARM is built upon a unique feature which makes it a very popular architecture to be used in embedded applications. Its design is very simple in compare to many other general-purpose processors in the sense that it can be manufactured with using a small number of transistors. Therefore the die size becomes very small, leaving plenty of space for custom ASIC[6] designs.

A highly notable architectural advantage of ARM is its low power consumption, since the aim of both ARM ISA and the pipeline design are minimum power consumption (which is very important in the design of the satellite). Lower energy consumption means less heat would be generated and also battery life would be maximized.

Another important feature of ARM is that it is highly modular: *"The only mandatory component of an ARM processor is the integer pipeline; all other components, including caches, MMU, floating point and other co-processors are optional, which gives a lot of flexibility in building application-specific ARM-based processors"* [35]. The high performance/power ratio of an ARM processor is another important feature. According to Intel, the XScale processor family can provide the performance of a Pentium II processor while using fifty times less energy[7]. [35]

2.4.4.6 Thumb concept

Thumb has been created as an extension to the ARM architecture. The main idea behind the creation of the Thumb was to reduce the size of the instruction sets of the ARM processor. In order to achieve this, two sets of instructions were implemented in the design:

- Standard 32-bit ARM instruction set
- 16-bit Thumb sets

In total, Thumb contains 36 instruction formats which are the re-coded 16-bit op-codes of the 32-bit ARM instruction set. These new 16-bit op-codes are decompressed to their ARM instruction set on execution. This allows the system bus to be only 16-bits wide, which reduces power consumption and area. By keeping the 32-bit instruction sets of ARM separate from Thumb instructions, the decoding logic becomes extremely simple which means small area is used and the ARM performance and low power concept is also maintained. By using Thumb-aware cores, the developer can write performance-critical codes in ARM and size-critical routines in Thumb. The code density improvement is around 30% [32] in compare to ARM, which brings Thumb-aware processors below CISC processors in terms of code size [32].

[6] Application Specific Integrated Circuits
[7] PXA255 XScale at 400 MHz; Intel Pentium II at 300 MHz.

2.4.4.7 Jazelle-enhanced processors

Hardware implementation of the Java virtual machine was done in some ARM processors which provides the ability to execute Java code as a third execution state along with the ARM and Thumb modes.

2.4.4.8 ARM 7 core specifications

ARM 7 is ideally suited for embedded systems which require low power consumption in addition to the RISC performance. The small die size makes it much easier to include a RAM, ROM, DSP, etc. inside the micro-controller package. Main core specifications of the ARM 7 are as follows [2]:

- 32-bit RISC address space
- 0.6 mA/MHz at 3 volt power supply operation
- Fast interrupt response mechanism
- 17-25 MIPS at 25 MHz / 3 volt operation
- 0.8 micron CMOS process
- Virtual memory support

2.4.4.9 ARM7

The ARM7 architecture is based on the RISC processor architecture principle. This processor offers very low power consumption along with high performance. It contains eleven basic instruction types. There are total of 37 registers implemented in the design: 31 general registers (32-bit) and 6 status registers. There are 16 general registers (R0 - R15) and one or two status registers available. The 16 registers can be accessed directly at any mode. All registers except R15 (Program Counter (PC)) are general purpose registers and can be used for data storage. The address bus is 32-bit wide. In order for ARM7 to have backwards compatibility with earlier processors, it can be configured to use a 26-bit address bus. This design is now considered obsolete. ARM7 block diagram is illustrated in Figure 2.1.

2.4.4.10 ARM7TDMI

The ARM7TDMI[8] is a 32-bit RISC processor with 3-stage pipelining Von Neumann load/store architecture. The processor contains the two ARM and Thumb instruction sets. There are 31 32-bit general registers implemented in the design along with 6 status registers which makes a total of 37 registers. The number of available registers differs depending on the state the processor is in. For example in the ARM state, 16 general register (R0 - R15) and one status register (R16) are available. The R16 is the Current

[8]T: Supports Thumb instruction set, D: Contains debug extentions, M: Enhanced 32×8 multiplier, I: EmbeddedICE macrocell.

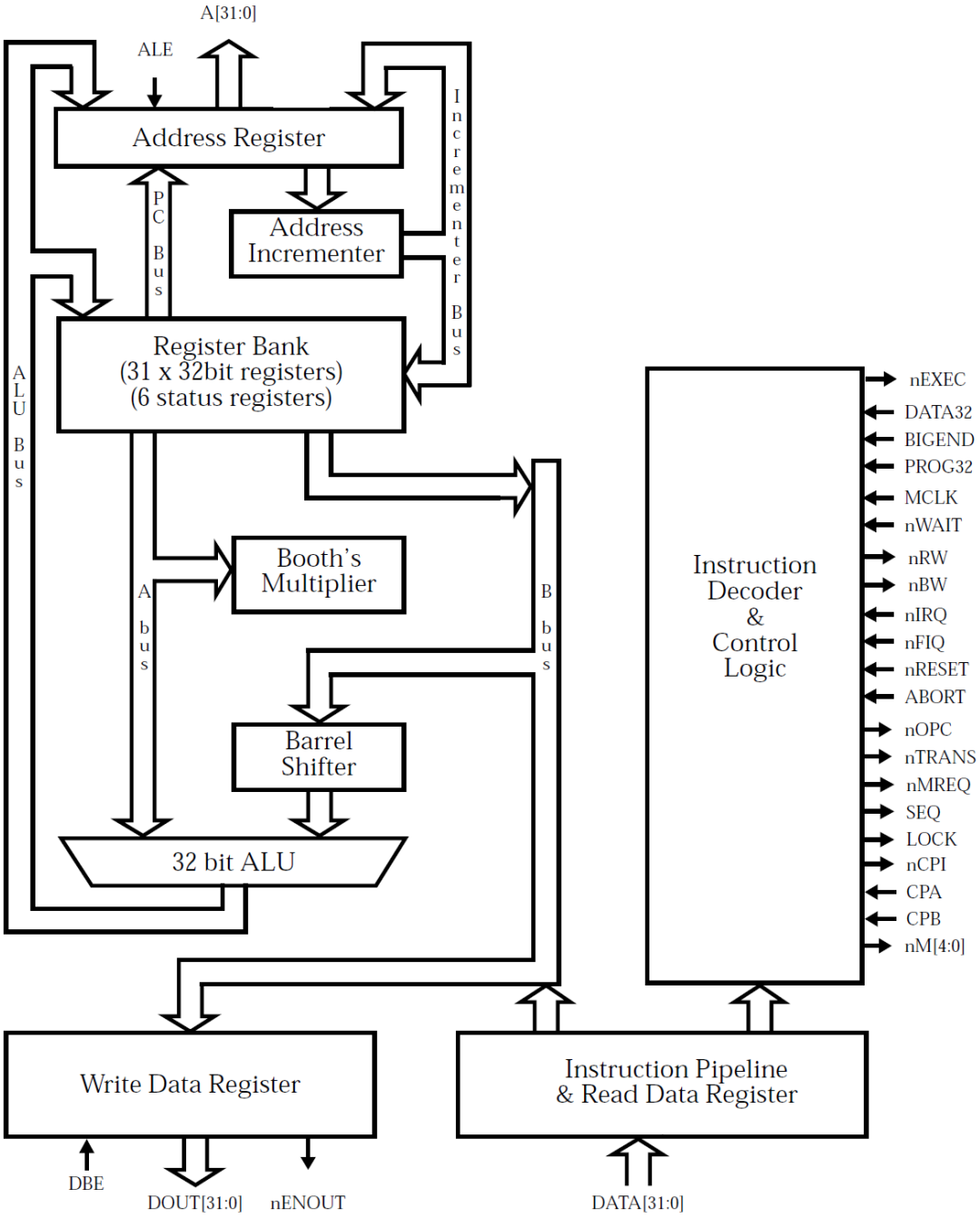


Figure 2.1: ARM7 Block Diagram [2]

Program State Register (CPSR) which contains code flags and mode bits, indicating the state the processor is in. The Program Counter (PC) is stored in R15 [6]. This architecture contains a single data and address bus and the instructions can operate on 8-, 16- and 32-bit data types. The following is the characteristics of this processor

architecture:

- 32/16-bit RISC architecture
- 32-bit ARM instruction set to achieve maximum performance
- 16-bit Thumb instruction set for increased code density
- Unified bus interface, 32-bit data bus carries both instructions and data
- 3-stage pipeline
- 32-bit ALU
- 32×8 Multiplier
- Very small die size and low power consumption
- Coprocessor interface

2.4.4.11 ARM9TDMI

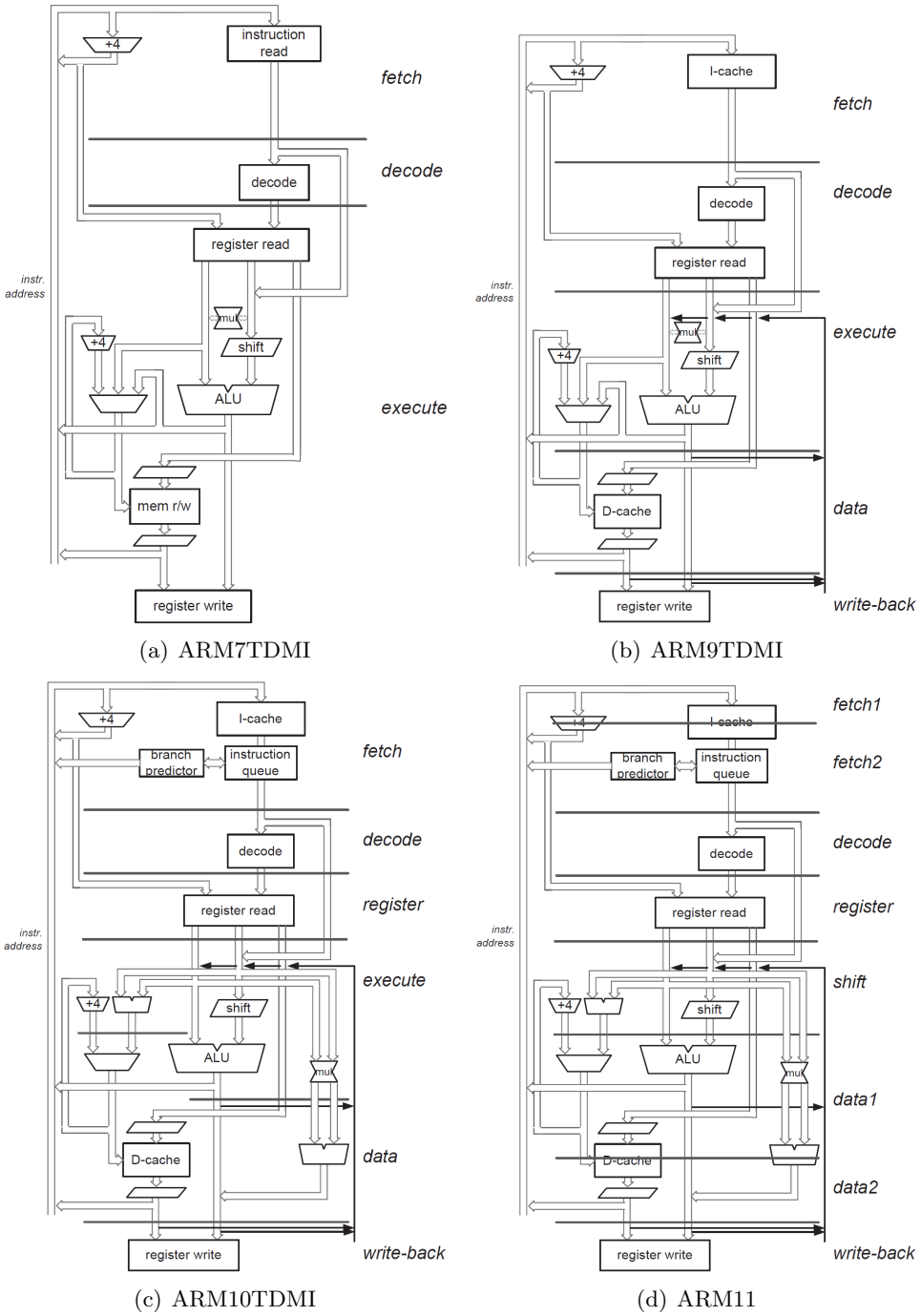
The ARM9TDMI architecture design has 16-bit Thumb implemented in it. The following is the features of the ARM9:

- 32-bit RISC processor with ARM and Thumb instruction sets
- 5-stage integer pipeline achieves 1.1 MIPS/MHz
- Up to 300 MIPS (Dhrystone 2.1) in a typical 0.13μm process
- Single 32-bit AMBA[9] bus interface
- MMU[10] supporting Windows CE, Symbian OS, Linux and Palm OS
- Integrated instruction and data caches
- Excellent debug support for SoC designers, including ETM interface
- 8-entry write buffer - avoids stalling the processor when writes to external memory are performed
- Portable to latest 0.18μm, 0.15μm, 0.13μm silicon processes.

2.4.4.12 ARM Pipelining technologies

As it was mentioned before, one of the important characteristics of the ARM processor architecture is the simplicity of the design. In this section a brief description of the ARM pipeline technologies is provided starting from the 3-stage pipelining to the 8-stage pipelining [35].

[9] Advanced Microcontroller Bus Architecture
[10] Memory Management Unit



(a) ARM7TDMI (b) ARM9TDMI

(c) ARM10TDMI (d) ARM11

Figure 2.2: ARM pipelining evolution [35]

1. **3-stage pipelining**

   The RISC architecture has a 5-stage pipelining: fetch, decode, execute, memory access, register write back. The original design of the ARM pipeline which is the

same from ARM1 until ARM7TDMI is the 3-stage fetch-decode-execute pipeline which completes one instruction in one cycle in the absence of memory access. Figure 2.2(a) shows the 3-stage pipelining in ARM7TDMI.

2. **5-stage pipelining**

   In 3-stage pipelining, availability of only one memory port is assumed which results in a stall at every data transfer execution. This is because the memory is being read from or written to, therefore the next instruction cannot be fetched. To avoid the stall, a modification was done which is currently being used in ARM9TDMI (shown in Figure 2.2(b)).

3. **6-stage pipelining**

   More improvements were made to the pipelining with ARM10 core (Figure 2.2(c)). In this design, both data and instruction bus have become 64-bit wide. This allows two instructions to be fetched at each cycle. This enabled the branch prediction unit to look ahead the instruction stream and as a result, eliminating penalty of branches for loops being executed many times. Also the 64-bit bus allows transfer of two registers at a time which improves the multiple-register transfer instruction performance. For the memory access stage to become faster, because of simplicity of address calculation, another adder is implemented which completes its calculation in less than one cycle. This provides one and a half cycle for memory access.

4. **8-stage pipelining**

   The 8-stage pipelining was introduced in ARM11 (Figure 2.2(d)). It contains two main modifications. A separate stage is dedicated for the shift operation. Access to both instruction and data cache is distributed across 2 pipeline stages. Fetch and decode stages are still executed in order. But the execution stage is split into three separate pipelines.

2.5 CISC and RISC comparison

This section is provided to clarify and summarize the main differences between CISC- and RISC-based processor architectures.

The advantages of using a processor developed with a RISC design implemented in it are much more than using the same processor designed and implemented with CISC concept. CISC is a Complex Instruction Set Computer. It contains several instructions with different modes which makes it complicated to use for a simple application. In RISC-based architecture, it is tried to reduce the number of instructions and simplify them. This is accomplished by the new instruction sets designed for this architecture. Due to the pipelining process of RISC, some commands may take one or two more cycles to finish execution, since they have to pass through a gate array.

Another benefit of using RISC architecture would be the fact that it requires as little memory referencing as possible. Typically, numerous on-chip registers are available for programmers to use. Arithmetic operations are done on the registers instead of directly

Table 2.2: CISC and RISC comparison

CISC	**RISC**
Complex instructions taking multiple cycles	Simple instructions taking one cycle
Any instruction may reference memory	Only LOAD/STORE reference memory
Not pipelined or less pipelined	Highly pipelined
Instructions interpreted by the microprogram	Instructions executed by the hardware
Variable format instructions	Fixed format instructions
Many instructions and modes	Few instructions and modes
Complex in the microcode	Complex in the compiler
Single register set	Multiple register set

on memory locations. Processor performance would be increased by not operating directly on memory and by including fast static RAM to split the data and instruction caching.

To decrease software complexity on RISC, the instruction set is executed directly on the hardware instead of using the microcode ROM in the processor. Therefore there will be no code translation between the instructions and the hardware. Table 2.5 shows the comparison and differences between the CISC and RISC architectures.

To sum up, the following would be the benefits of using a RISC-based microcontroller (instead of one based on CISC) in an embedded system such as the Delfi-n3Xt nanosatellite:

- Much faster response to non-deterministic events
  Interrupt response of the processor is improved by removing the instructions that take many cycles to finish execution. Higher sampling rate for real-time events can be taken by the use of these smaller instructions. This is critical since this satellite is a real-time system.

- Simpler assembler coding
  Since the instructions length and addressing modes on the RISC architecture are much less than the ones on CISC, it is possible to have a much more simple code instructions and therefore it is easier to debug the assembly code.

- High throughput
  As mentioned above, since there are a few instructions to decode in a RISC machine, the decoding would be simpler and faster than a CISC machine (which requires twice the number of cycles required in RISC). This leads to a higher throughput in RISC processors.

- Access to memory devices
  In a CISC machine, if the data values that have been read from the memory are required again, an attempt to access the memory or cache is made again, while in a RISC machine, because of the number of the registers used in the architecture, an access to the registers will quickly provide the required values.

- Low power consumption
  Nowadays, RISC-based microcontrollers, have very low power consumption. Some 8-bit or 16-bit microcontrollers are currently available in the market which use up to 2mW/MHz. This is a quality which is very attractive for embedded systems.

2.6 Memories

One of the tasks of Delfi-n3Xt nanosatellite is to gather data while in orbit. A part of this data is the housekeeping information generated by on-bard subsystems. Therefore sufficient permanant data storage capacity is very important. Before the data is written into the permanent memory by the microcontroller, it is gathered in RAM (Random Access Memory). Therefore it is important to have enough RAM available.

Memory storages are divided based on their core characteristics. These characteristics are volatility, mutability, accessibility and addressability.

2.6.1 Volatility

This category is created to clarify the method used to maintain data in memory.

2.6.1.1 Volatile Memory

This type of memory requires power in order to retain the stored data inside it. SRAM (Static Random Access Memory) and DRAM (Dynamic Access Memory) are two most known examples of this memory type.

DRAM stores every data bit in a capacitor inside the RAM chip. Because of the capacitor charge leakage, the value will eventually change. To prevent this, the capacitor is re-charged (refreshed) periodically. Because of the need for this refresh, this type of memory is called *Dynamic*.

SRAM (unlike DRAM) does not require the periodic refresh. In total six transistors are required for each storage cell. Four transistors are required to store the data. Two more transistors are used as access controls during read/write operations.

The advantage of dynamic memory is its simplicity. For every bit, one transistor and one capacitor are required, where in static memory, six transistors are needed. DRAM is mainly used as the large memory in workstations, personal computers and game-consoles (e.g. Microsoft XBox 360, Nintendo Wii, Sony Playstation). SRAM is used for other parts such as cache.

2.6.1.2 Non-Volatile Memory

This type of memory does not require constant power to maintain stored data inside it. Flash memory, HDD (Hard Disk Drive), optical drives and holographic memory are a few examples of this type of memory.

2.6.2 Mutability

This category is based on the read/write abilities provided by the memory. There are three types of memory in this category.

2.6.2.1 Mutable Memory

In this type of memory, data can be overwritten at any time. This type of memory is typically used as the primary storage in computers. RAM (Random Access Memory) and HDD (Hard Disk Drive) are examples of this memory type.

2.6.2.2 Immutable Memory

This type of memory, allows the data to be written in it one time. After the data is written, it cannot be modified.

2.6.2.3 Slow Write, Fast Read Memory

This type of memory allows data to be read and written multiple times. However, the writing speed is much slower then the reading speed. Flash memory is an example of this type.

2.6.3 Accessibility

There are two types of memory in this category. This category is based on the access time between two memory cells.

2.6.3.1 Random Access

In this memory type, the access time for every location in the memory is approximately equal. RAM, flash memory and all sorts of SSD (Solid State Drive) are examples of this type.

2.6.3.2 Sequential Access

Each cell in this type of memory is read in serial order. Access time of a memory cell in this type, depends on the position of the last read memory cell. Tapes are examples of this type of memory.

2.6.4 Addressability

Memories in this category differ in their addressing methods. There are three types of memory in this category.

2.6.4.1 Location Addressable

In this type of memory, each memory cell is associated a memory address. This address is numerical. To perform operation on a cell, each cell is selected with its address. This type of addressing is not human-readable.

2.6.4.2 File Addressable

In this type, the data is divided into variable length files. Every file is given a human readable path and name. This type of memory is based on a location addressable memory. The operating system that reads/writes on this type of memory, provides the file system abstraction.

2.6.4.3 Content Addressable

In this memory type, each individual data unit is selected based on the basis of the content stored in that unit. Unlike the location addressable memory (where user supplies an address and the memory returns data in that address), a data word is supplied to this type of memory by user. The memory searches to find the supplied data word. If the data is found, the memory returns the address(es) of the location(s) where the data was found. This memory can be generated using either hardware or software. CPU cache is an example of hardware content addressable memory.

2.7 Microcontroller Units (MCU)

A microcontroller unit is an integrated circuit, consisting of a central processing unit (CPU) as its core, with a set of memory storage and peripherals around that core. A microcontroller can be used in embedded systems. Inside a microcontroller (aside from its CPU), memory (Flash, ROM, RAM), timers, I/O ports and other device controllers can be found.

The processor of a microcontroller can be any type of a processor architecture. A few examples of the CPU are Snapdragon (ARM-based), Motorola 68000, Atmel AT91SAM, Atmel AVR (RISC-based), Texas Instruments (TI) MSP430 (RISC-based) and Microchip PIC (RISC-based).

Microcontrollers usually contain general purpose input/output (GPIO) pins. GPIO can be configured by program to act either as input or output. When a pin is set as input, external signals (e.g. sensors) can be read by the microcontroller. When a pin is set as output, it can produce signals to derive external devices (e.g. speaker, motor, LED, etc.).

Nowadays, the common peripherals designed in microcontrollers are: analog to digital converter (ADC), digital to analog converter (DAC), pulse width modulator (PWM), real-time timer (RTT), real-time clock (RTC), universal asynchronous receiver/transmitter (UART).

2.8 Summary

Delfi-n3Xt nanosatellite requires a microcontroller with data storage, to collect and maintain the housekeeping data. In this chapter, a background on processor design and concepts were briefly explained. Instruction Set Architecture (ISA) and microcode were described, followed by CISC and RISC architecture concepts and their main differences.

Different types of available memories were introduced and a general description on what a microcontroller is made of was given.

3 Evaluation Methodology

This chapter presents the evaluation methodology used in this thesis. In order to find the most suitable microcontroller for Delfi-n3Xt nanosatellite, a comparison is done between the available microcontrollers in the market. In order to make this comparison, commercial off-the-shelf microcontrollers will be selected. Afterwards, they will be compared to each other and get ranked. The result of this ranking system will determine the most suitable microcontroller for this satellite.

3.1 Ranking system

In order to make a trade off between the compared microcontrollers, a ranking system is proposed. In this system, the microcontrollers are compared with each other at each criterion. The analysis criteria is explained in section 3.2. The result of this system, is a table which contains the rank of each microcontroller. The following subsections explain different parts of this system.

3.1.1 Point

Every microcontroller will be given points during the comparison. This is based on how well it fulfills the compared criterion. Points range from zero to five. Zero point means that the MCU does not provide any support for the compared criterion. Five points is given to a microcontroller if it completely meets the criterion requirements. The points in between are given depending on the difference between the microcontrollers in that criterion.

3.1.2 Weight

Every comparison criterion is given a *weight*. The weight is an integer number ranging between 1 and 5. It indicates the importance of that criterion. A criterion with very high importance is associated with a weight of 5. Lower weight is associated with a criterion with less importance.

3.1.3 Score

Each microcontroller gains its score for each criterion by multiplying the points received by the microcontroller (in that specific comparison) to the criterion weight. For example, if a microcontroller gains 3 points for its power consumption, the weight of the criterion (in this case 5) is multiplied by the points and a score of 15 is associated with the microcontroller in that criterion. These scores are used in the final comparison.

3.1.4 Final Result

The final result of this system is a table. This table shows the points, weights and total score of all the microcontrollers. The final score for each microcontroller is the sum of all the scores gained by that MCU. The microcontroller with highest score, is the most suitable microcontroller for the Delfi-n3Xt nanosatellite.

3.2 Microcontroller Selection and Analysis Criteria

In order to make a comparison, a list of microcontrollers has to be created. To make this list, commercial off-the-shelf microcontrollers are chosen. However, making a list of all microcontrollers that have been successfully used in different applications, will be quite long. Therefore, microcontrollers with the previously described architectures will be chosen for comparison. In the following chapter, a discussion is made on how the microcontrollers are selected as candidates for comparison.

By going through this list and examining the requirements of Delfi-n3Xt mission, some of the microcontrollers will be eliminated. These are the microcontrollers which do not meet one or more items in the requirement list. The remaining microcontrollers will be compared with each other. Based on the ranking system described in section 3.1, the most suitable microcontroller for this satellite will be chosen.

The following subsections provide detailed information on each requirement item. Each item is a criterion used in the ranking system. An explanation is provided for each item to clarify why it is required.

3.2.1 Processing Power (MIPS)

MIPS stands for Million Instructions Per Second and is used for representing the processor speed. It indicates how many instructions can be completed within one second. Processor speed depends on many factors such as the instruction type and execution order. Instruction execution rates differ in different processors and in different clock frequencies. An instruction may take several clock cycles to complete. Also, several independent instructions could be executed together. Therefore, benchmarking programs such as Whetstone, Dhrystone and CoreMark were developed to calculate the processor speed.

Microcontroller manufacturers measure the processor speed and include it (in terms of MIPS) in the datasheets. Microcontroller chosen for this satellite should at least provide the minimum processing power required by the subsystems. The OBC has to have high enough performance to manage all the subsystems. Task of ADCS is to calculate the precise position and trajectory of this satellite. This is one of the subsystems that needs to process the data fast. The higher the the microcontroller performance, the faster it can process and provide data. However, this is dependent on the task given to the microcontroller. Since the performance is of high importance, it is given the highest weight.

Weight = 5

3.2.2 Power Consumption per MIPS

The electrical power supply subsystem of this satellite (EPS) can generate at most 20 watts. This energy has to be distributed among all satellite subsystems. Based on the components that can be used in different subsystems, the power consumption of a subsystem can become as high as 16 watts. Delfi-n3Xt Power Budget [37] covers the information on how much power is dedicated to each subsystem in different modes. In order to have the lowest possible power consumption on every subsystem, it is essential for the microcontroller to have a very low power consumption. Power consumption is also one of the very important criteria of this satellite. Therefore, the highest weight is associated with it.

Weight = 5

3.2.3 I^2C Hardware Support

I^2C stands for Inter-Integrated Circuit. It is a protocol mainly used for connecting peripherals to motherboards or embedded systems. The I^2C bus uses two bidirectional lines named SDA (Serial Data Line) and SCL (Serial Clock Line) for communication. TWI (Two Wire Interface) is an alternative name used for I^2C. In this protocol, one master device is selected and all other devices become the slaves. To begin data transmission, first, the master sets the clock (SCL) which is used for data transfer synchronization. Afterwards, the master selects a slave (slave address is transmitted on SDA) and after the target slave responds, data transmission begins (on SDA). At this point, data could be transmitted either from master to slave or the other way around. After the data transmission is over, the communication will stop when the sender sends the stop bit on the data line.

The standard communication between all the subsystems of this satellite, is the I^2C protocol. It would be a great advantage if the microcontroller supports this bus interface directly in its hardware. Otherwise, the bus has to be either implemented in software or by using an external I^2C bridge. Adding I^2C bridge in every subsystem requires an additional space on each subsystem board. Coding this protocol will take additional space in the microcontroller storage as well as its RAM. Therefore it is preferred if the microcontroller supports it in its hardware. This bus is one of the very important interfaces and the highest weight is associated with it.

Weight = 5

3.2.4 Serial Peripheral Interface (SPI)

SPI bus is a synchronous serial communication bus used for communications with devices and peripherals that support SPI. This interface typically uses four lines for communication: SCLK (Serial Clock), MOSI/SIMO (Master Output, Slave Input), MISO/SOMI (Master Input, Slave Output) and SS (Slave Select). In this protocol, one master node is selected and all other nodes become the slaves. Each node is associated with an address. To begin communication, first, the master configures the clock (SCLK). The clock speed should not exceed the maximum clock frequency supported by the target slave node. Then it selects the target slave device by pulling its Slave Select signal to

low. A full-duplex data transmission begins and data is transferred on MOSI and MISO lines.

This satellite requires the SPI communication for storage and communication with a few peripherals such as Flash memory storage and gyroscope. Many Flash memory storage devices use SPI for communication. Nowadays, most of the microcontroller manufacturers implement an SPI peripheral in the microcontroller hardware. If this interface is not implemented in the microcontroller, it has to be either coded or implemented in the required subsystems by using a SPI to I/O bridge. The code will use storage space in microcontroller along with four I/O pins. A bridge takes up additional space on the subsystem circuit board. Since this peripheral is important, it is associated with a high weight.

Weight = 4

3.2.5 Analog to Digital Converter (ADC)

Some of the devices on the subsystems of this satellite, produce analog data. This data has to be processed by the microcontroller. Since processors process digital data, the analog data has to be converted to digital. This conversion is done by an ADC interface. To reduce the area usage on a subsystem board, it is preferred to have an ADC interface implemented in the hardware of the microcontroller. Otherwise, external ADC has to be implemented on the subsystem board. Monitoring the power supply and reading analog sensors are examples of analog data process in this satellite. This interface is one of the important requirements and it is associated with a high weight.

Weight = 4

3.2.6 Digital to Analog Converter (DAC)

A few devices in this satellite (e.g. reaction wheel) require analog signal to operate properly. A processor produces digital signals. These signals can be converted to analog by using a DAC. Therefore a DAC circuit is required to operate each analog device on this satellite. It is preferred to have a DAC interface implemented in the microcontroller. Otherwise an external DAC bridge has to be implemented on subsystem boards that operate analog instruments. This interface is also one of the important requirements and a high weight is associated with it.

Weight = 4

3.2.7 Memory Type

Today's microcontrollers have either Flash, ROM or both implemented as their program storage unit. The main difference between a microcontroller with Flash storage and one with ROM storage, is that the ROM memory based (EPROM) microcontroller is a one-time memory chip. This means that the data stored in it would not be modified after the satellite has been lunched. On the other hand, with a Flash based memory (a specific type of EEPROM) the flight software can be modified and an update feature can be added to this satellite.

With today's growing Flash technology, most of the microcontrollers have embedded Flash as their main program memory. Using Flash based microcontrollers is recommended instead of using ROM. In case gathered data needs to be stored inside the microcontroller (instead of an external storage), Flash based memory provides this ability. In this case, the microcontroller will be able to store this data in a section of the microcontroller memory which can be read later. This is an important requirement and a high weight is associated with it.

Weight = 4

3.2.8 Watchdog Timer (WDT)

A watchdog timer is required in order to provide a system hardware reset. In case the OBC, PTRX, ADCS or any other subsystem module faces failure and is not responding (*hanged*), the watchdog timer will restart the hanged system. This circuit has to be implemented in the design. Today's microcontrollers have it implemented in them. In case this timer is not implemented inside a microcontroller, it has to be provided on the subsystem boards to ensure continues operation of the microcontroller. This is one of the very important requirements and the highest weight is associated with it.

Weight = 5

3.2.9 Brownout Reset (BOR)

Brown-out reset is a circuit that restarts the microcontroller in case of a power interrupt. This prevents the processor to operate in a power condition less than what it is designed for. It is done by completely cutting the power from microcontroller and restating it. This circuit is required to assure the microcontroller will not malfunction due to power interrupts. If this circuit is not present inside a microcontroller, it can be implemented on the subsystem board. This circuit is not of high importance, since EPS distributes the power throughout the subsystems properly. However, it is better to prevent damage to a microcontroller caused by power interruption by using this circuit.

Weight = 3

3.2.10 Temperature Range

Sun is the major source of heat in space. Temperature of the satellite, varies based on its position in relation to Sun. Since the satellite will be orbiting Earth, it goes in Earth shadow and comes out. The surface of the satellite facing the sun will become very warm and when it goes in the shadow it becomes very cold. The microcontroller has to be able to stand the extreme temperatures. According to the thermal analysis result of Delfi-C^3 [25], the temperature range inside the satellite for normal operation remained within -10°C and 25°C. Therefore the microcontroller must be able to operate within this temperature range. In case a microcontroller does not support this range, it should be kept within its operational range by designing and using a thermal management system This system has to keep the microcontroller temperature within its supported operating range. This is an important requirement and a high weight is associated with it.

Weight = 4

3.2.11 Space Mission Past Experience

A microcontroller that has not been tested for space applications, has a chance of failure during its mission. To reduce this chance, past space mission experiences of microcontrollers are checked. In case a microcontroller has been used in other successful space missions before, it will receive additional points. This criteria is important and a high weight is associated with it.

Weight = 4

3.2.12 Packaging

Different manufacturers create microcontrollers in different package types, material and sizes. A microcontroller packaged as QFP (Quad Flat Package) is easier to mount and solder on the PCB than the QFN (Quad Flat No leads) or MLF (Micro Lead Frame) packages. These packages are categorized under SMD (Surface Mount Devices) packaging. It is of high importance for the microcontroller package to be of SMD type, due to the hight limitation between the satellite subsystem boards. This is an important requirement and a high weight is associated with it.

Weight = 4

3.2.13 Development Environment

A good development environment speeds up the software development and debugging of microcontrollers. It is important to have a powerful tool for debugging the hardware and software. Generally, for every microcontroller, there is one or more programming utilities available in the market. It is important to have a debugging tool available either next to the programming software or implemented in it. Therefore, the tools get ranked based on their features and provided licensing. Since a good development environment is important, a high weight is associated with it.

Weight = 4

3.3 Summary

In this chapter, the ranking system used for comparison and analysis of microcontrollers has been described. In this system, definition of weight, points and scores have been explained. Furthermore, all the requirements regarding the microcontroller and satellite have been described in detail. Each requirement is a comparison criterion in the ranking system. Each criterion has been associated with a weight based on its importance. The final result of the ranking system, is a table which contains the final score of all microcontrollers. The microcontroller with the highest score is the most suitable unit for this satellite.

4 Results

In chapter 3, method for finding the most suitable microcontroller for Delfi-n3Xt nanosatellite was described. These methods are applied and explained in this chapter. The trade-off begins with choosing candidate microcontrollers. The main characteristics and features of these microcontrollers are explained in detail. Afterwards, the comparison and trade-off between them is provided. This comparison is based on the requirements of the Delfi-n3Xt nanosatellite. To show the trade-off clearly, there will be tables created for each criteria. These tables hold points received by each microcontroller in every criteria compared. Once all criteria have been covered, the ranking system (as explained in chapter 3) will be applied and based on its result, the most suitable microcontroller will be selected.

4.1 Microcontroller Candidates

Satellites are very expensive and normally it takes several years to build them. As long as they are on the ground in the lab, they can be tested for survival in space under different conditions and circumstances. In case of malfunction (considering the cost and the amount of time spent on repairing parts of the satellite in situe or remotely), in many cases building a new satellite would be more reasonable than repairing it while in space. Therefore the chosen components for the satellite must be highly reliable in order to have a successful mission.

For Delfi-n3Xt nanosatellite, the components are chosen to be commercial off-the-shelf products. Usually for this purpose, components that have already been in space are selected. Since the satellite development is a long process (takes several years), the components that have already been used and tested in space become old and meanwhile new components with new technologies are designed. These components can provide more peripherals (in the same package size) and higher performance along with lower power consumption than their predecessors. But since they are not tested for space applications, they may not be fully reliable.

There is a large number of microcontrollers available in the market. Each of these microcontrollers have been created to serve applications such as mobile phones, portable medical devices, multi-media players, measurement control systems, etc. Companies that build these microcontrollers, also specify the field they can be used in. This makes it possible to decide which microcontrollers not to consider in an application, by taking a quick look. AMD, Atmel, Intel, Microchip, Motorola, NXP, Samsung, Texas Instruments (TI) and Toshiba, are several well known companies providing these microcontrollers.

To make a trade-off between microcontrollers, the selection is made within the ones using either CISC or RISC concept. Within the CISC-based microcontrollers, the ones based on Motorola's 68000 architecture are chosen. Within RISC-based MCUs, ARM

based microcontrollers are chosen for trade-off, along with the MSP430-based architecture developed by Texas Instruments. MSP430-based architecture and ARM-based MCUs are chosen because of their low power consumption and their field of applications. The MSP430 microcontrollers have been designed for portable and low power medical applications and control and monitoring systems. Atmel's ARM-based microcontrollers (more specifically AT91SAM[1] series), are designed for low power and portable multi-media applications. Motorola's 68000-based microcontrollers (DragonBall series) are used in low power multi-media devices and PDAs[2].

Before using any microcontroller, it is necessary to have an understanding of its hardware and software requirements. Therefore, the following subsections are provided to cover detailed information on these microcontrollers.

4.1.1 Motorola MC68328 CISC MCU

The MC68328 series of microcontrollers are 32-bit low power microcontroller based on the Motorola 68000 CISC processor. This series contains three microcontrollers. These microcontrollers are described in the following subsections.

4.1.1.1 DragonBall EZ (MC68EZ328)

Figure 4.1 shows the functional block diagram of this microcontroller. Its Processor core (68EC000) is the updated implementation of 32-bit M68000 architecture. Main features of this core are [23]:

- Low power, static HCMOS (High-speed Complementary Metal-Oxide-Semiconductor)
- 16-bit data bus, 32-bit address bus
- Sixteen 32-bit address and data registers
- 56 instruction types, 14 addressing modes and five main data types
- Seven priority levels for interrupt control

This microcontroller supports up to 45 parallel GPIO (General Purpose I/O) pins. These pins can either be configured as general purpose I/O or dedicated peripheral interface. Each pin is independently programmable as GPIO even when the other pins are used as peripherals dedicated pins. [23]

The 16-bit timer supports various modes. Its input clock frequency can be programmed (driven from system clock). It can act as a counter and trigger an external interrupt (or event) when it reaches a preset value. It can also be set to count external events. [23]

The PWM (Pulse Width Modulator) can be used to generate custom pulse signals. It has a 5-byte FIFO (First In First Out) implemented in it which allows the CPU to process other interrupts while PWM is being supplied by data. [23]

[1] SAM: Smart ARM-based Microcontrollers
[2] Personal Digital Assistant

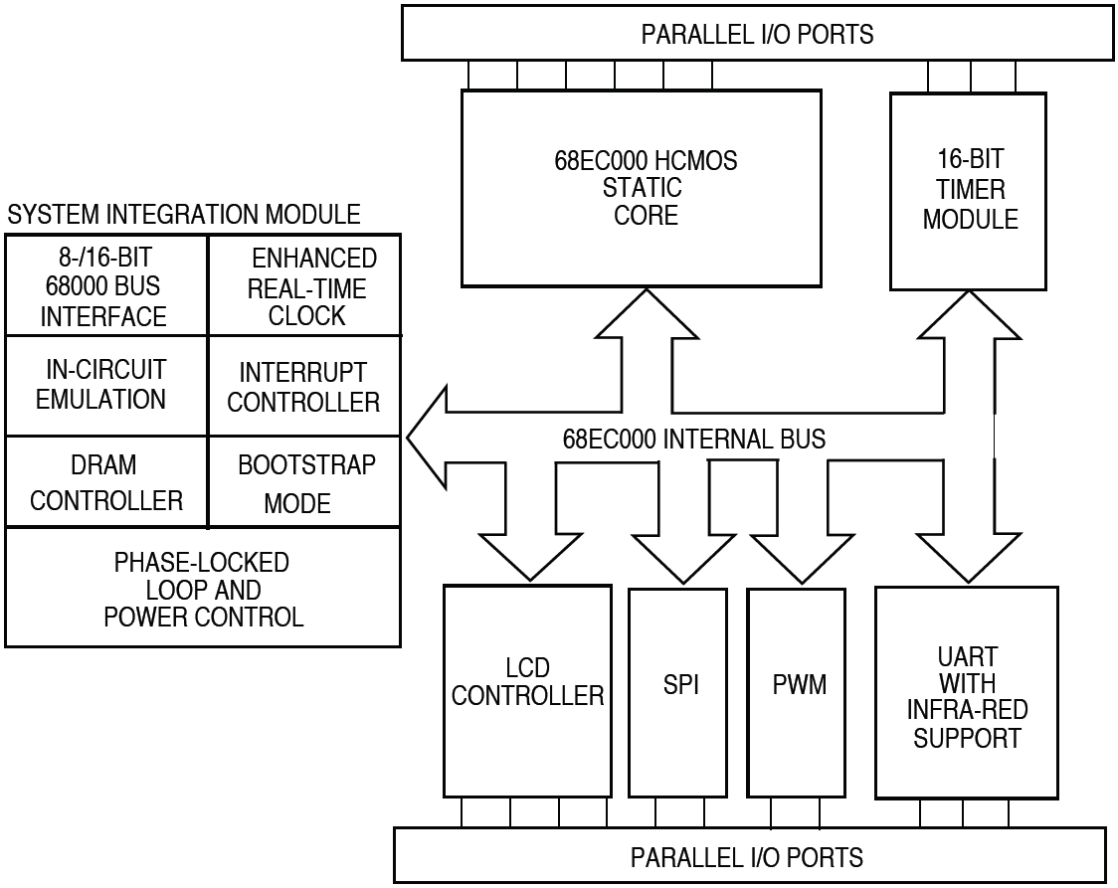


Figure 4.1: MC68EZ328 Block Diagram [23]

The SPI (Serial Peripheral Interface) only supports master mode. It means that the SPI can initiate the connection and control a device connected to it. [23]

This controller also supports UART (Universal Asynchronous Receiver/Transmitter) and can communicate with external devices through a standard asynchronous protocol. The interface communicates at baud-rates between 300 bps (bits per second) to 1152 kbps (kilo-bits per second). The IrDA (Infra-red Data Association) transceiver is driven directly by UART which provides the pulses. [23]

The LCD (Liquid Crystal Display) controller can display data on LCD modules. The display data is fetched from memory. Control signals, clocks, frame line pulse and data to the LCD module, are provided by this interface. It supports monochrome STN (Super-Twisted Nematic) LCD modules of maximum sixteen gray level with frame rate control. System RAM is used for display memory. [23]

The enhanced RTC (Real-Time Clock) provides time-of-day with one second resolution by using a crystal oscillator (either 32.768 kHz or 38.4 kHz) as its clock source. The system WDT (Watchdog Timer) is implemented in this module. It uses the 1 Hz clock inside the real-time clock to generate an interrupt or reset signal to system every two seconds. [23]

DRAM (Dynamic Random Access Memory) controller provides an interface to connect most DRAM chips. It supports maximum two DRAM banks of maximum 4 MB (mega bytes) each. [23]

The internal Phased-Locked Loop (PLL) unit on this chip provides clock signal for the clock synthesizer, by either connecting an external crystal or an external oscillator. The power control unit can turn off peripherals that are not being used to save power. It can also reduce processor clock speed or disable the processor altogether. The interrupt controller module runs during the low-power mode. An interrupt provided by this module, allows the chip to wake up from this mode. These interrupts can be programmed interrupts, peripheral interrupts or from the real-time clock. [23]

All interrupts (whether internal or external) are prioritized by the interrupt controller module. The following are the interrupt controller features: [23]

- Fully nested interrupt environment
- Prioritized interrupts
- Programmable vector generation
- Unique vector number generated for each interrupt level
- Interrupt/Wakeup masking

Bootstrap mode provides debugging and monitoring on the chip. A program (or data) can be downloaded to the system RAM via UART. Once downloaded, it can be executed, which provides a simple debugging environment (useful for failure analysis) and to update the programs stored in the Flash memory. The following are the bootstrap mode features: [23]

- Allows system initialization and data download to system memory via UART
- Accepts execution commands used for running programs in system memory
- 8-byte long instruction buffer to store and execute 68000 instruction

In-circuit emulation (ICE) module can be used to support a low-cost emulation on this microcontroller. It consists of one breakpoint insertion unit (for breakpoint instruction), an interrupt controller (level 7 interrupt), one address signal comparator and one control signal comparator with masking (support single or multiple hardware execution/bus breakpoints). [23]

To summarize, the following is the characteristics of this microcontroller [23]:

- Performance up to 2.7 MIPS
- Maximum clock frequency: 16.58 MHz
- 3.3 V Single supply
- Maximum current consumption: 20 mA

- 45 GPIO pins
- No ADC or DAC
- PWM (Pulse Width Modulator) controller
- Real-Time Clock (RTC)
- SPI (Master support)
- No I^2C
- 16-bit Timer
- Watchdog Timer
- LCD Controller
- In-Circuit Emulator

Table 4.1 shows the characteristics of this microcontroller, with regards to the evaluation criteria.

Performance	**Power Consumption per MIPS**	**I^2C**	**SPI**	**ADC**	**DAC**
2.7 MIPS	66 mW / 2.7 MIPS = 24.4	-	M	-	-

Memory	**WDT**	**BOR**	**Temperature**	**Packaging**	**Dev. Env.**
60 KB	Yes	-	0 to +70 °C	TQFP, PBGA	EASy68K

Table 4.1: DragonBall EZ Criteria Fulfillment
(M: Master, TQFP: Thin Quad Flat Package, PBGA: Plastic Ball Grid Array)

4.1.1.2 DragonBall VZ (MC68VZ328)

Figure 4.2 shows the block diagram of this microcontroller. Its processor core is named FLX68000. It is identical to the MC68EC000 microprocessor used in DragonBall EZ [21]. Main features of this core are [21]:

- Low power, static HCMOS
- 16-bit data bus, 32-bit address bus
- Sixteen 32-bit address and data registers
- 56 instruction types, 14 addressing modes and 5 main data types
- Seven priority levels for interrupt control

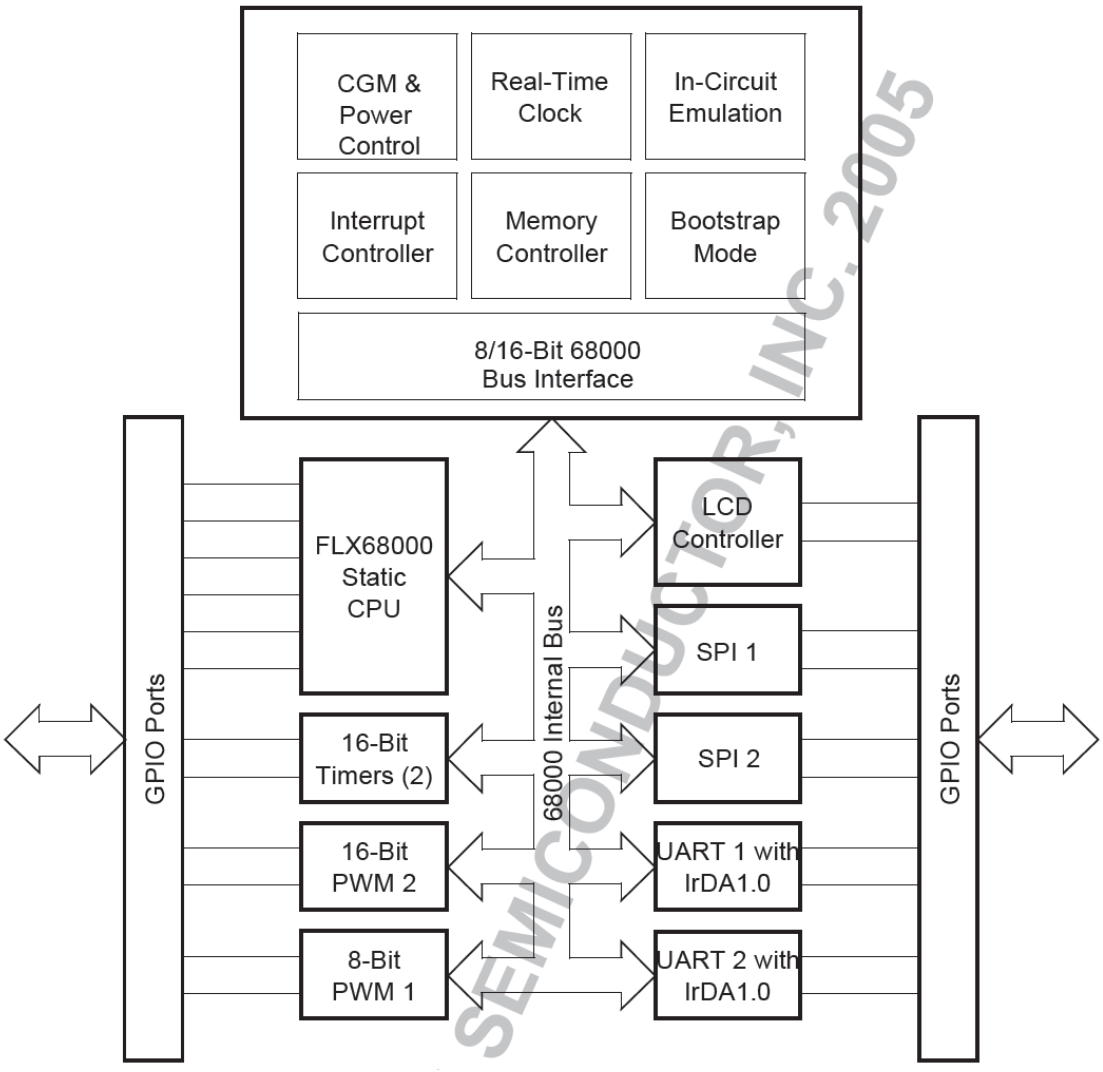


Figure 4.2: MC68VZ328 Block Diagram [21]

This microcontroller supports up to 76 parallel GPIO pins. These pins have been grouped together in 10 ports (A-G, J, K and M). They can either be configured as general purpose I/O or dedicated peripheral interface. Each pin is independently programmable as GPIO, even when the other pins are used as peripherals dedicated pins. [21]

This microcontroller contains two 16-bit timers. They can be used in various modes. They can act as a counter and trigger an external interrupt when they reach a preset value. They can also be set to count external events. These two timers can be cascaded together as one 32-bit timer. [21]

There are two PWMs available on this microcontroller. Each of them has three operation modes: 1. playback; 2. tone; 3. digital to analog conversion (DAC). They can be used to generate high-quality digital sounds, custom pulse signals and analog waves. The 8-bit PWM contains a 5-byte FIFO which allows the CPU to process other

interrupts while PWM is being supplied by data. This is done by reducing the number of interrupts the CPU receives. [21]

Two SPI modules (SPI 1 and SPI 2) are implemented in this microcontroller. SPI 2 can only work in master mode. It means that it can initiate the data transfer and control a device connected to it. The other SPI module (SPI 1) can be configured as either master or slave. [21]

This controller provides supports for two UART ports. They can be used to communicate with external serial devices through a standard asynchronous protocol. UART 1 is the same as the UART implemented in DragonBall EZ. UART 2 is the enhanced version of UART 1. In this module, the receive and transmit FIFOs (RxFIFO and TxFIFO) have been enlarged. This results in reduced number of software interrupts. An improvement to both UARTs, is the system clock frequency input which supports 33.16 MHz (two times the DragonBall EZ clock frequency). [21]

The LCD controller in this microcontroller is the improved version of DragonBall EZ LCD controller. It contains a new algorithm to improve flickering effect. In addition to 4-, 2- and 1-bit wide LCD data bus supported by DragonBall EZ LCD controller, it supports 8-bit wide LCD data bus. [21]

The RTC module provides time-of-day with one-second resolution by using crystal oscillator (either 32.768 kHz or 38.4 kHz) as its clock source. The system WDT (Watchdog Timer) is implemented in this module. It uses the 1 Hz clock inside the real-time clock to generate an interrupt or a system reset every two seconds. [21]

DRAM controller provides an interface to connect either 8-bit or 16-bit DRAM. It supports EDO (Extended Data Output) RAM, Fast Page Mode and synchronous DRAM. It supports maximum two DRAM banks (16 MB $\times$ 16 or 32 MB $\times$ 8 DRAM or SDRAM[3]). [21]

For the internal Clock Generation Module (CGM), a stable clock source is provided by the clock synthesizer which can operate either on an external crystal or an external oscillator. The output clock frequency of CGM, can be adjusted by writing to the CGM frequency select register. This module can be disabled to save power. The power control module can turn off peripherals that are not being used to save power. It can also control CPU cycles to optimize power consumption. This modules supports three (normal, doze and sleep) power-saving modes. In sleep mode, the CGM wakes up automatically when an unmasked interrupt (either internal or external) occurs. [21]

All interrupts (whether internal or external) are prioritized by the interrupt controller module. The following are the interrupt controller features: [21]

- Fully nested interrupt environment
- Prioritized interrupts
- Programmable vector generation
- Unique vector number generated for each interrupt level

Bootstrap mode provides debugging and monitoring on the chip. A program (or data) can be downloaded to the system RAM via either UART 1 or UART 2 controllers.

[3]Synchronous Dynamic Random Access Memory

Once downloaded, it can be executed, which provides a simple debugging environment (for failure analysis) and to update the programs stored in the Flash memory. The following are the bootstrap mode features: [21]

- Allows system initialization and data (and program) download to system memory via UART 1 or UART 2
- Accepts execution commands used for running programs in system memory
- 32-byte instruction buffer to store and execute 68000 instruction

In-circuit emulation (ICE) module is designed to support a low-cost emulation on this microcontroller. It consists of one breakpoint insertion unit (for breakpoint instruction), an interrupt controller (level 7 interrupt), one address signal comparator and one control signal comparator with masking (support single or multiple hardware execution/bus breakpoints). [21]

To summarize, the following is the characteristics of this microcontroller [21]:

- Performance up to 5.4 MIPS
- Maximum clock frequency: 33 MHz
- 3.3 V Single supply
- Maximum current consumption: 40 mA
- 76 GPIO pins
- No ADC or DAC
- PWM (Pulse Width Modulator) controller
- Real-Time Clock (RTC)
- SPI (Master/Slave support)
- No I^2C
- 2x 16-bit Timers
- Watchdog Timer
- LCD Controller
- In-Circuit Emulator

Table 4.2 shows the characteristics of this microcontroller, with regards to the evaluation criteria.

Performance	Power Consumption per MIPS	I^2C	SPI	ADC	DAC
5.4 MIPS	132 mW / 5.4 MIPS = 24.4	-	M/S	-	-

Memory	WDT	BOR	Temperature	Packaging	Dev. Env.
60 KB	Yes	-	0 to +70 °C	TQFP, MAPBGA	EASy68K

Table 4.2: DragonBall VZ Criteria Fulfillment
(M: Master, S: Slave, TQFP: Thin Quad Flat Package, MAPBGA: Mold Array Process-Ball Grid Array)

4.1.1.3 DragonBall Super VZ (MC68SZ328)

Figure 4.3 shows the block diagram of this microcontroller. This microcontroller has the same FLX68000 core used in DragonBall VZ implemented in it. As mentioned before, this core is identical to the MC68EC000 microprocessor used in DragonBall EZ [22]. Main features of this core are [22]:

- Up to 66.32 MHz clock
- Fully compatible with MC68000 and MC68EC000
- 16-bit data bus, 32-bit address bus
- Sixteen 32-bit address and data registers
- 56 instruction types, 14 addressing modes and 5 main data types
- Seven priority levels for interrupt control

There are a total number of 93 GPIO pins in 12 groups. All the ports are multiplexed with peripherals. In addition, they are integrated with dedicated internal pull-up and pull-down resistors. Pins can be configured as general purpose I/O or dedicated peripheral interface. Each pin is independently programmable as GPIO, even while the other pins are used as peripherals dedicated pins. [22]

The two identical 16-bit timers implemented in this microcontroller can provide a 15 ns resolution (at 66.32 MHz system clock). They can be used in various modes. They can act as a counter and trigger an external interrupt when they reach a preset value. They can also be set to count external events. These two timers can be cascaded together as one 32-bit timer. [22]

There are two pulse width modulators (PWMs) available on this microcontroller. Each has three operation modes: 1. playback; 2. tone; 3. digital to analog conversion (DAC). They can be used to generate high-quality digital sounds, custom pulse signals and analog waves. The 8-bit PWM (PWM 1) contains a 5-byte FIFO which allows the CPU to process other interrupts while PWM is being supplied by data. This is done by reducing the number of interrupts the CPU receives. The 16-bit PWM (PWM 2) does not have a FIFO. [22]

The RTC (Real-Time Clock) module provides time-of-day with one-second resolution by using a 32.768 kHz clock input. It has a sampling timer with selectable clock frequency (512 Hz, 256 Hz, 64 Hz, 32 Hz, 16 Hz, 8 Hz, 4 Hz). The system WDT (Watchdog Timer)

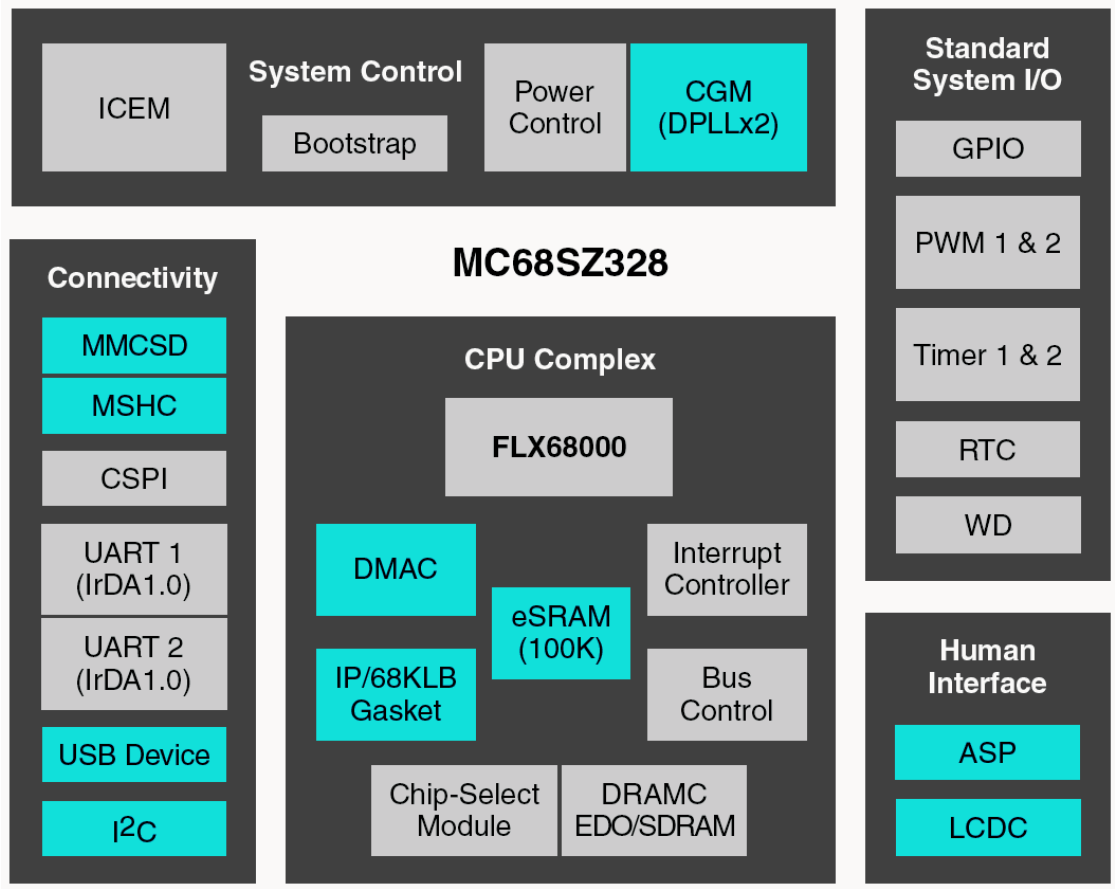


Figure 4.3: MC68SZ328 Block Diagram [22]

is implemented in this module. It uses the 1 Hz clock to generate either an interrupt or a system reset every two seconds. [22]

This microcontroller contains a single configurable serial peripheral interface (CSPI) module. In contrast to DragonBall VZ (which contains SPI1 and SPI2), this chip contains only one CSPI, which is an enhanced version of SPI 1. This module can be configured to operate as master or slave with DMA support. [22]

There are two identical UART modules implemented in this chip, which are nearly the same as UART 2 module used in DragonBall VZ. The main improvement in these modules would be the difference in FIFO size and DMA support. They feature 32-byte FIFO for transmit and 32-byte FIFO for receive. They provide baud rates of 600 bps to 460.8 kbps. [22]

The LCD controller in this microcontroller supports monochrome/color LCD panels with up to 640×480 resolution with external memory support. It also contains an 8-bit PWM to control LCD contrast by software. It supports 16-bit wide LCD data bus in addition to 8-, 4-, 2- and 1-bit wide LCD data bus. [22]

DRAM controller (DRAMC) provides an interface to connect EDO (Extended Data Output) DRAM and SDRAM (synchronous DRAM). The chip select signals can be pro-

grammed together as either EDO DRAM controller or an SDRAM controller. Therefore EDO DRAM and SDRAM cannot co-exist. It supports maximum four banks (active simultaneously) per SDRAM chip select (single data rate of 256 Mbit, 128 Mbit, 64 Mbit). [22]

The Chip-Select Module equips this microcontroller with twelve chip-select signals: ten general purpose, programmable chip-select signals (used to select external devices), one signal to select eSRAM and one for the emulation module. [22]

In the internal Clock Generation Module (CGM), there are two oscillators used in conjunction with a multiplier/divider chain. These clock signals are used throughout the MC68SZ328 processor. This module generates all clock signals including CPU, DMA and USB clock. The programmable PLL allows clock generation of up to 200 MHz from a 16 MHz clock input. [22]

The Power Control module, controls the efficiency of the clocking system to optimize the power consumption of this microcontroller. This module provides four power modes: normal, burst, doze and sleep. PCM is turned off in normal mode. When it is enabled, the processor enters burst mode. If the clock burst width is reduced to zero, the clock is disabled and the processor enters the doze mode. When the chip enters sleep mode, all clocks (except the 32.768 kHz clock) are disabled. The real-time clock works on the 32.768 kHz clock and it is kept operational. [22]

All internal and external interrupts are supported and prioritized by the interrupt controller module. There are 7 interrupt levels (level 7 has the highest priority and level 1 has the lowest). This module features a fully nested interrupt environment and programmable vector generation. A unique vector number is generated for each interrupt level. [22]

Bootstrap mode provides debugging and monitoring on the chip. A program (or data) can be downloaded to the system RAM via either UART 1 or UART 2 controllers. Once downloaded, it can be executed, which provides a simple debugging environment (for failure analysis) and to update the programs stored in the Flash memory. The following are the bootstrap mode features: [22]

- Allows system initialization and data (and program) download to system memory via UART 1 or UART 2
- Accepts execution commands used for running programs in system memory
- 32-byte instruction buffer to store and execute 68000 instruction

In-circuit emulation (ICE) module is designed to support a low-cost emulation on this microcontroller. It consists of one breakpoint insertion unit (for breakpoint instruction), an interrupt controller (level 7 interrupt), one address signal comparator and one control signal comparator with masking (support single or multiple hardware execution/bus breakpoints). [22]

Multimedia card (MMC) is a seven-pin Flash based mass storage serial interface. A secure digital (SD) memory card is an enhanced multimedia card which uses two additional pins. The MMC/SD host controller implemented in this chip, supports both MMC and SD memory cards. It acts as the host device and provides command interrupter, protocol handler, error detection, clock control features and an application interface. [22]

There is a Memory Stick host controller (MSHC) implemented in this chip. It provides support for standard Memory Stick interface. There are 8-byte FIFO buffers for receive and transmit in this module. The bus clock frequency can be set up to 66.32 MHz. DMA is supported in this module which improves performance. [22]

This chip has an integrated USB device controller. It complies with USB version 1.1 with maximum speed of 12 Mbps. In a USB connection, there is only one host on the bus which is the bus master. The USB controller on this chip acts as a bus slave and can only communicate with a USB host. Therefore it does not generate any bus traffic, but only responds to requests made by the host. There are registers available to the programmer to enable/disable this module and to control traffic flow through the module. [22]

An I^2C module is implemented in this chip. The following are the key features of this module [22]:

- Support for 7-bit address
- Support for 3.0 V devices (up to 3.3 V)
- Multiple-master operation
- Software selectable acknowledge bit
- Interrupt driven byte-by-byte data transfer
- Start and Stop signals and acknowledge bit generation/detection
- Repeated start signal generation
- Bus-busy detection
- Calling address identification interrupt
- Arbitration-lost interrupt with automatic mode switching from master to slave

Analog Signal Processing (ASP) module, provides conversion capabilities for a variety of analog devices. There are several analog to digital converters (ADC) with 16-bit resolution (8-bit accuracy) available in this module. They can be used with any analog device including touch panel and pen. [22]

There is an embedded SRAM (eSRAM) module implemented in this chip. It has a total size of 100 kB (kilo bytes). It can be selected with a signals from the chip-select module and can be used as the display memory. [22]

Direct Memory Access Controller (DMAC), provides DMA capabilities between the memories (internal and external) and the following modules: MSHC, UART 1, UART 2, USB, MMCSD, CSPI and ADC. There is a 64-byte data FIFO (used as buffer for holding the transfer data during a DMA burst cycle) and a logic to generate interrupts and handshaking signals with a USB host. [22]

To summarize, the following is the characteristics of this microcontroller [22]:

- Performance up to 10.8 MIPS

- Maximum clock frequency: 66 MHz
- 3.3 V Single supply
- Maximum current consumption: 80 mA
- 93 GPIO pins
- 16-bit ADC
- no DAC
- 8-bit and 16-bit PWM controllers
- Real-Time Clock (RTC)
- SPI (Master/Slave support)
- I^2C (Multi-Master and Slave mode support)
- 2x 16-bit Timers
- Watchdog Timer
- LCD Controller
- USB Device
- Multimedia Card/Secure Digital (MC/SD) controller
- In-Circuit Emulator

Table 4.3 shows the characteristics of this microcontroller, with regards to the evaluation criteria.

Performance	**Power Consumption per MIPS**	**I^2C**	**SPI**	**ADC**	**DAC**
10.8 MIPS	264 mW / 10.8 MIPS = 24.4	MM/S	M/S	16-bit	-

Memory	**WDT**	**BOR**	**Temperature**	**Packaging**	**Dev. Env.**
60 KB	Yes	-	0 to +70 °C	MAPBGA	EASy68K

Table 4.3: DragonBall Super VZ Criteria Fulfillment
(M: Master, MM: Multi-Master, S: Slave, MAPBGA: Mold Array Process-Ball Grid Array)

4.1.2 TI MSP430 RISC MCU

The MSP430 MCUs are ultra low power microcontrollers with a 16-bit RISC-based processor implemented in them. They have been designed as mixed-signal processors and contain various peripherals along with a flexible clock system. The MSP430 MCUs are divided into multiple families based on their peripherals, amount of memory and the

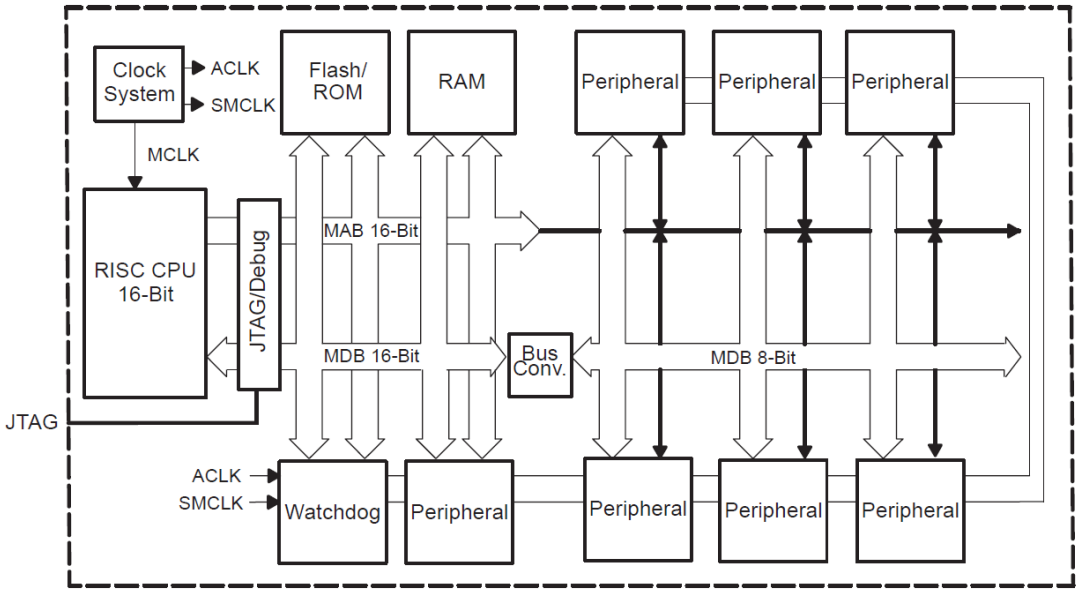


Figure 4.4: MSP430 Microcontroller General Functional Block Diagram [40]

application they can be used in. Figure 4.4 shows the general functional block diagram of all MSP430 microcontrollers.

There are many microcontrollers developed by TI in each family that contain the MSP430 architecture. All of them include and support JTAG interface for programming and run-time debugging. The microcontrollers in each family have some differences. The main difference between the families is in their performance and some of their peripherals. Therefore, the following sub-sections are provided to cover information about microcontroller series in each family.

4.1.2.1 MSP430F16x Series

This series of microcontrollers, use the 16-bit MSP430 RISC-based processor. This processor uses the von Neumann architecture. Figure 4.5 illustrates the architecture of MSP430F16x microcontroller series. Main features of this core are [44]:

- Low power consumption
- 16-bit address bus, 16-bit and 8-bit data bus
- Total 16 registers

  R0 to R3: Program Counter, Stack Pointer, Status Register and Constant Generator

  R4 to R15: General-Purpose Registers
- 51 instructions, three formats and 7 addressing modes

These microcontrollers provide up to 48 GPIO pins (six 8-bit I/O ports). The pins can be configured as general purpose I/O or dedicated peripheral interface. Each I/O pin is independently programmable as GPIO even when other pins are used as dedicated peripheral. Peripherals and the processor are connected together through address, data and control buses. [44]

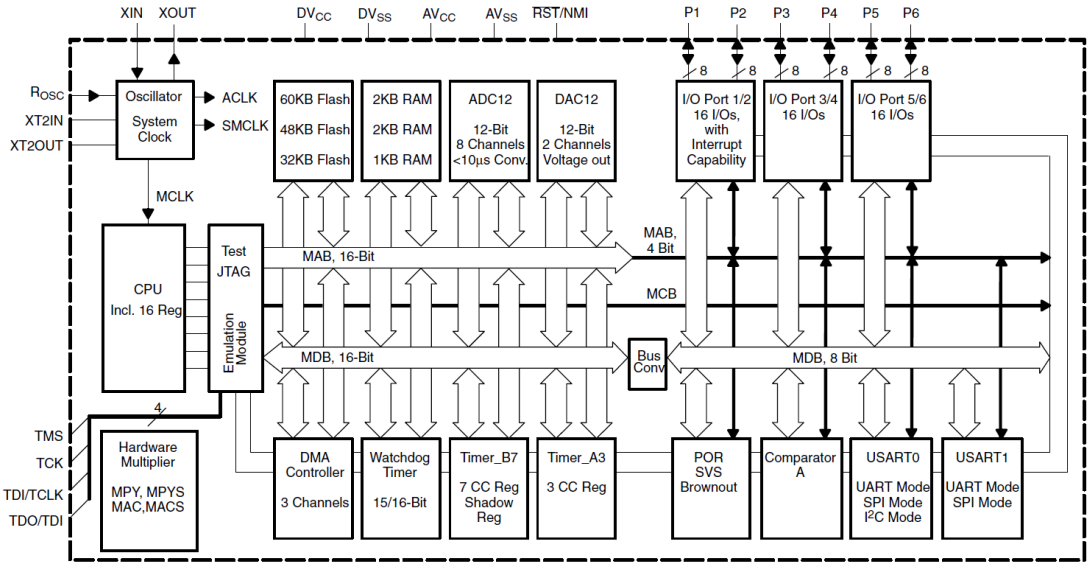


Figure 4.5: MSP430F16x Block Diagram [44]

These microcontrollers have a dedicated multiplier module implemented in them. This module can perform 16×16, 16×8, 8×16 and 8×8 bit operations. After the operands have been loaded in its registers, the result of its operations can be accessed immediately (no additional clock cycles required). [44]

The DMA controller provides data movement without processor intervention. The DMA reduces the power consumption as well as increasing the throughput of peripherals. Reduced power consumption of the system is done by allowing the CPU to remain in sleep mode while DMA moves data from memory to peripheral and vice versa. [44]

A watchdog timer (WDT) module is implemented in these microcontrollers to perform a controlled system restart after a software problem occurs. This timer is clocked by an auxiliary 32.768 kHz oscillator. If this timer is not required in an application, it can be configured as an interval timer. The watchdog timer counter is a 16-bit up counter and is only accessible by the watchdog timer control register. [44]

There are two 16-bit timers implemented in these microcontrollers: Timer A3 and Timer B7. Timer A3 is a timer/counter with 3 capture/compare registers. Timer B7 is a timer/counter with 7 capture/compare registers. They both support PWM output and interval timing. Timer interrupts can be generated from the counter on overflow condition and from each capture/compare register. [44]

The ADC12 module, is a 12-bit SAR (Successive Approximate Register) analog to digital converter. It has 8 input channels, all individually configurable. There are 16

conversion-result storage registers. It is possible to select either internal or external voltage reference by software. The on-chip (internal) reference voltage generation is also software selectable (1.5 V or 2.5 V). [44]

The DAC12 module, is a 12-bit voltage output digital to analog converter. This module has 2 output channels. The output voltage resolution can be configured in 8-bit or 12-bit mode. Voltage reference can be selected from either internal or external reference. [44]

There are two USART [4] hardware modules implemented in these chips. They provide support for UART, SPI and I^2C. USART0 supports all three modes. USART1 only provides support for UART and SPI interfaces. There is a baud-rate generator implemented in the USART modules. Baud-rate can be set by configuring USART control registers. The maximum UART baud-rate supported, is one-third of the UART source clock frequency. [44]

The SPI mode on both USART modules, supports both master and slave modes. Transmit and receive registers are separate 8-bit registers. Individual interrupts are generated when module is ready to transmit, or when it has received data. [44]

The following are the main features of the I^2C module [44]:

- Byte/word format transfer
- 7-bit and 10-bit device addressing modes
- Built in FIFO for read and write buffer
- 16-bit wide data access
- Programmable clock generation
- Start bit detection (slave mode) for auto-wake up from low power mode
- Multi-master and slave mode operation
- Standard mode up to 100 kbps, fast mode up to 400 kbps

The comparator module, is an analog voltage comparator. It supports supply voltage supervision, monitoring external analog signals and precision slope analog to digital conversions. One of the examples of using this comparator is for temperature readings. [44]

Both power-on reset (POR) and brown-out reset (BOR) circuits are available here. POR signal is generated when either powering up the chip, restarting the chip (from reset pin) or when there is an SVS[5] low condition. BOR circuit detects low supply voltages and resets the chip by triggering the POR signal. The supply voltage supervisor module, monitors the supply voltage. It can be configured to provide a POR or set a flag. SVS is disabled after a brownout reset occurs to conserve current consumption. [44]

[4] Universal Synchronous/Asynchronous Receiver/Transmitter
[5] Supply Voltage Supervisor

JTAG[6] module and Bootstrap loader (BSL) are implemented in these microcontrollers. The Flash memory can be programmed using either one of these modules. The JTAG interface port is protected with a fuse. Once this fuse is blown, it is completely disabled and further access to the chip is not possible via JTAG. The Bootstrap loader enables reading and programming the Flash memory or RAM using a UART interface. A 256-bit, user defined password, protects access to the Flash memory via BSL.

In addition to the above programming methods, the CPU of these microcontrollers, is able to write to its Flash memory. In this method, the user (programmer) can provide data through any available means (SPI, UART, etc.). The developed software can receive the data and program the Flash memory. This is a completely customizable programming solution and can be used for updating the program memory. [44]

To summarize, the following is the characteristics of this microcontroller [44]:

- Performance up to 8 MIPS
- Maximum clock frequency: 8 MHz
- 250 μA / MIPS in active state
- 1.8 - 3.6 V
- Maximum current consumption: 2.640 mA
- 48 GPIO pins
- 12-bit ADC Successive Approximative Register (SAR)
- 12-bit DAC
- SPI (Master/Slave support)
- I^2C (Multi-Master and Slave mode support)
- 2x 16-bit Timers
- Watchdog Timer
- Hardware multiplier
- Analog Comparator
- Brownout Reset
- JTAG Interface
- No LCD Controller
- No USB support

Table 4.4 shows the characteristics of this microcontroller, with regards to the evaluation criteria.

[6]Joint Test Action Group, is used for testing printed circuit boards (PCB) using boundary scan. Boundary scan is a method to test interconnects in an integrated circuit or a PCB.

Performance	Power Consumption per MIPS	I^2C	SPI	ADC	DAC
8 MIPS	5.808 mW / 8 MIPS = 0.726	MM/S	M/S	12-bit	12-bit

Memory	WDT	BOR	Temperature	Packaging	Dev. Env.
60 KB	Yes	Yes	-40 to +85 °C	LQFP, QFN	CCE, CCS, CrossWorks

Table 4.4: MSP430F16x Criteria Fulfillment
(M: Master, MM: Multi-Master, S: Slave, LQFP: Low-profile Quad Flat Package, QFN: Quad Flat No-Lead)

4.1.2.2 MSP430x261x Series

This series of microcontrollers, use the 16-bit MSP430 RISC-based processor. This processor uses the von Neumann architecture. Figure 4.6 illustrates the functional block diagram of MSP430F261x microcontroller series. Main features of this core are [41]:

- Low power consumption
- 16-bit address bus, 16-bit and 8-bit data bus
- Total 16 registers

  R0 to R3: Program Counter, Stack Pointer, Status Register and Constant Generator

  R4 to R15: General-Purpose Registers
- 51 instructions, three formats and 7 addressing modes

These microcontrollers come in two 64-pin and 80-pin packages. The 64-pin packages provide up to 48 GPIO pins (six 8-bit I/O ports). The 80-pin packages provide up to 64 GPIO pins (eight 8-bit I/O ports). Diagram illustrated in Figure 4.6, shows MSP430F261x with eight I/O ports (80-pin package). The pins can be configured as general purpose I/O or dedicated peripheral interface. Each I/O pin is independently programmable as GPIO even when other pins are used as dedicated peripheral. Peripherals and the processor are connected together through address, data and control buses. [41]

These microcontrollers have a dedicated multiplier module implemented in them. This module can perform 16×16, 16×8, 8×16 and 8×8 bit operations. After the operands have been loaded in its registers, the result of its operations can be accessed immediately (no additional clock cycles required). [43][41]

The DMA controller provides data movement without processor intervention. The DMA reduces the power consumption as well as increasing the throughput of peripherals. Reduced power consumption of the system is done by allowing the CPU to remain in sleep mode while DMA moves data from memory to peripheral and vice versa. This module has three independently configurable channels. [43][41]

A watchdog timer+ (WDT+) module is implemented in these microcontrollers to perform a controlled system restart after a software problem occurs. This timer is clocked by an auxiliary 32.768 kHz oscillator. If this timer is not required in an application, it can

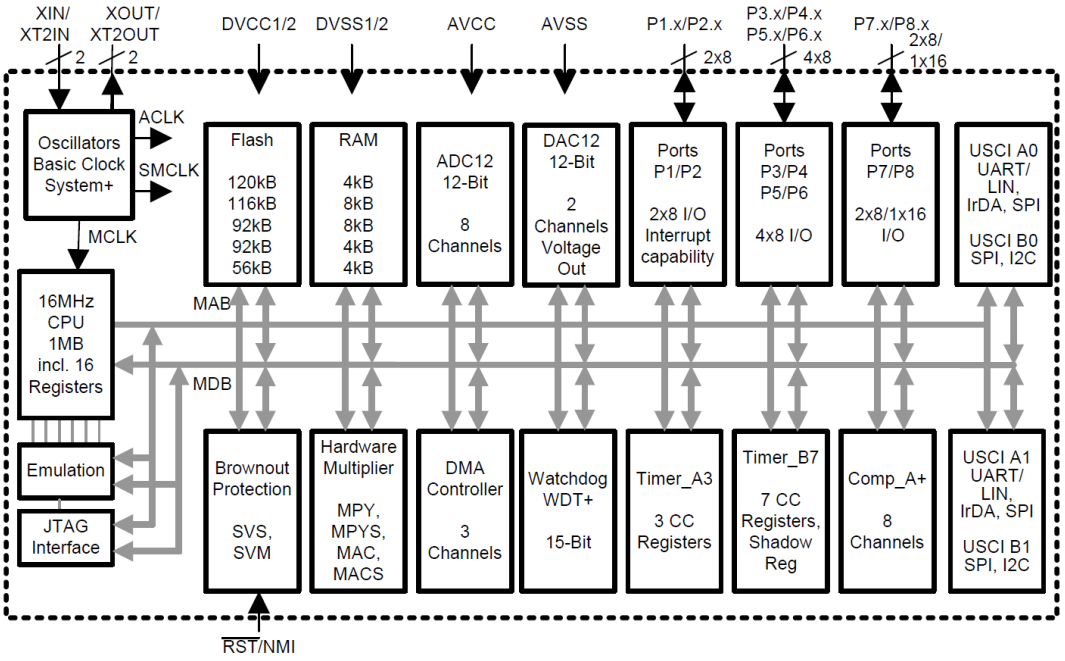


Figure 4.6: MSP430F261x Block Diagram [41]

be configured as an interval timer. Its counter is a 16-bit up counter and is only accessible by the watchdog timer+ control register. The watchdog timer+ module, provides a fail-safe clocking feature. It prevents disabling the WDT+ clock while this module is acting as a watchdog. If the auxiliary or secondary clock sourcing the watchdog timer+ fails, it automatically switches to the master clock. In case the master clock is sourced from a crystal and the crystal has failed, the fail-safe will activate the DCO[7] and will use it as the master clock source. [43][41]

There are two 16-bit timers implemented in these microcontrollers: Timer A3 and Timer B7. Timer A3 is a timer/counter with 3 capture/compare registers. Timer B7 is a timer/counter with 7 capture/compare registers. They both support PWM output and interval timing. Timer interrupts can be generated from the counter on overflow condition and from each capture/compare register. [43][41]

The ADC12 module, is a 12-bit SAR[8] analog to digital converter. It has 8 input channels, all individually configurable. There are 16 conversion-result storage registers. It is possible to select either internal or external voltage reference by software. The on-chip (internal) reference voltage generation is also software selectable (1.5 V or 2.5 V). [43][41]

The DAC12 module, is a 12-bit voltage output digital to analog converter. This module has 2 output channels. The output voltage resolution can be configured in 8-bit or 12-bit mode. Voltage reference can be selected from internal reference or external. [43][41]

[7]Digitally-Controller Oscillator
[8]Successive Approximate Register

There are two USCI[9] modules implemented in these chips. They provide support for serial communications. USCI_A0 and USCI_A1 modules support UART, IrDA, LIN[10] and SPI. The USCI_B0 and USCI_B1 modules support SPI and I^2C modes. [43][41]

In the UART mode, there are two modes (low-frequency and oversampling) for baud-rate generation. In low-frequency mode, baud-rates can be generated from low frequency clock sources (e.g. 9600 baud (bits per second) from a 32.768 kHz crystal). In the over-sampling mode, higher input clock frequencies are supported. The maximum generated baud-rate, is 256000 baud. [43][41]

The SPI mode on both USART modules, supports both master and slave modes. Transmit and receive registers are separate 8-bit registers. Data transmit and receive can be configured as either LSB-first (Least Significant Bit) or MSB-first (Most Significant Bit) transmission. Individual interrupts are generated when module is ready to transmit, or when it has received data. [43][41]

The following are the main features of the I^2C module [43][41]:

- Byte/word format transfer
- 7-bit and 10-bit device addressing modes
- Built in FIFO for read and write buffer
- 16-bit wide data access
- Programmable clock generation
- Start bit detection (slave mode) for auto-wake up from low power mode
- Multi-master and slave mode operation
- Standard mode up to 100 kbps, fast mode up to 400 kbps

The comparator module (Comparator_A+), is an analog voltage comparator. It supports supply voltage supervision, monitoring external analog signals and precision slope analog to digital conversions. One of the examples of using this comparator is for temperature readings. This comparator has an input short switch bit that can be set to build a simple sample-and-hold comparator. [43][41]

Both power-on reset (POR) and brown-out reset (BOR) circuits are available here. POR signal is generated when either powering up the chip, restarting the chip (from reset pin) or when there is an SVS[11] low condition. BOR circuit detects low supply voltages and resets the chip by triggering the POR signal. The supply voltage supervisor module, monitors the supply voltage. It can be configured to provide a POR or set a flag. SVS is disabled after a brownout reset occurs to conserve current consumption. [43][41]

JTAG[12] module and Bootstrap loader (BSL) are implemented in these microcontrollers. The Flash memory can be programmed using either one of these modules. The

[9] Universal Serial Communication Interface

[10] Local Interconnect Network

[11] Supply Voltage Supervisor

[12] Joint Test Action Group, is used for testing printed circuit boards (PCB) using boundary scan. Boundary scan is a method to test interconnects in an integrated circuit or a PCB.

JTAG interface port is protected with a fuse. Once this fuse is blown, it is completely disabled and further access to the chip is not possible via JTAG. The Bootstrap loader enables reading and programming the Flash memory or RAM using a UART interface. A 256-bit, user defined password, protects access to the Flash memory via BSL. [43][41]

In addition to the above programming methods, the CPU of these microcontrollers, is able to write to its Flash memory. In this method, the user (programmer) can provide data through any available means (SPI, UART, etc.). The developed software can receive the data and program the Flash memory. This is a completely customizable programming solution and can be used for updating the program memory. [43][41]

To summarize, the following is the characteristics of this microcontroller series [43][41]:

- Performance up to 16 MIPS
- Maximum clock frequency: 16 MHz
- 250 μA / MIPS in active state
- 1.8 - 3.6 V
- Maximum current consumption: 5.840 mA
- 64 GPIO pins
- 12-bit ADC Successive Approximative Register (SAR)
- 12-bit DAC
- SPI (Master/Slave support)
- I^2C (Multi-Master and Slave mode support)
- 2x 16-bit Timers
- Watchdog Timer
- Analog Comparator
- Brownout Reset
- JTAG Interface
- No LCD Controller
- No USB support
- No Hardware multiplier

Table 4.5 shows the characteristics of this microcontroller, with regards to the evaluation criteria.

Performance	Power Consumption per MIPS	I^2C	SPI	ADC	DAC
16 MIPS	12.848 mW / 16 MIPS = 0.803	MM/S	M/S	12-bit	12-bit

Memory	WDT	BOR	Temperature	Packaging	Dev. Env.
120 kB	Yes	Yes	-40 to +85 °C	LQFP, QFN	CCE, CCS, CrossWorks

Table 4.5: MSP430x261x Criteria Fulfillment
(M: Master, MM: Multi-Master, S: Slave, LQFP: Low-profile Quad Flat Package, QFN: Quad Flat No-Lead)

4.1.2.3 MSP430C337

This microcontroller, uses the 16-bit MSP430 RISC-based processor. This processor uses the von Neumann architecture. Figure 4.7 illustrates the functional block diagram of MSP430C337 microcontroller. Main features of this core are [38]:

- Low power consumption
- 16-bit address bus, 16-bit and 8-bit data bus
- Total 16 registers

  R0 to R3: Program Counter, Stack Pointer, Status Register and Constant Generator

  R4 to R15: General-Purpose Registers
- 27 instructions, three formats and 7 addressing modes

This microcontroller comes in a 100-pin package with up to 40 GPIO pins (five 8-bit I/O ports). The pins can be configured as general purpose I/O or dedicated peripheral interface. Each I/O pin is independently programmable as GPIO even when other pins are used as dedicated peripheral. Peripherals and the processor are connected together through address, data and control buses. [38]

A dedicated multiplier module is implemented in this microcontroller. It can perform 16×16, 16×8, 8×16 and 8×8 bit operations. After the operands have been loaded in its registers, the result of its operations can be accessed immediately (no additional clock cycles required). [39][38]

A watchdog timer (WDT) module is implemented in this chip to perform a controlled system restart after a software problem occurs. This timer is clocked by an auxiliary 32.768 kHz oscillator. If this timer is not required in an application, it can be configured as an interval timer. Its counter is a 15/16-bit up-counter and is only accessible by the watchdog timer control register. [39][38]

There is one 16-bit timer module implemented in this microcontrollers (Timer_A). It is a timer/counter with 5 capture/compare registers. It supports multiple simultaneous timings, multiple output waveforms (PWM output), multiple capture/compare and any combination of these. Timer interrupts can be generated from the counter on overflow condition and from each capture/compare register. [39][38]

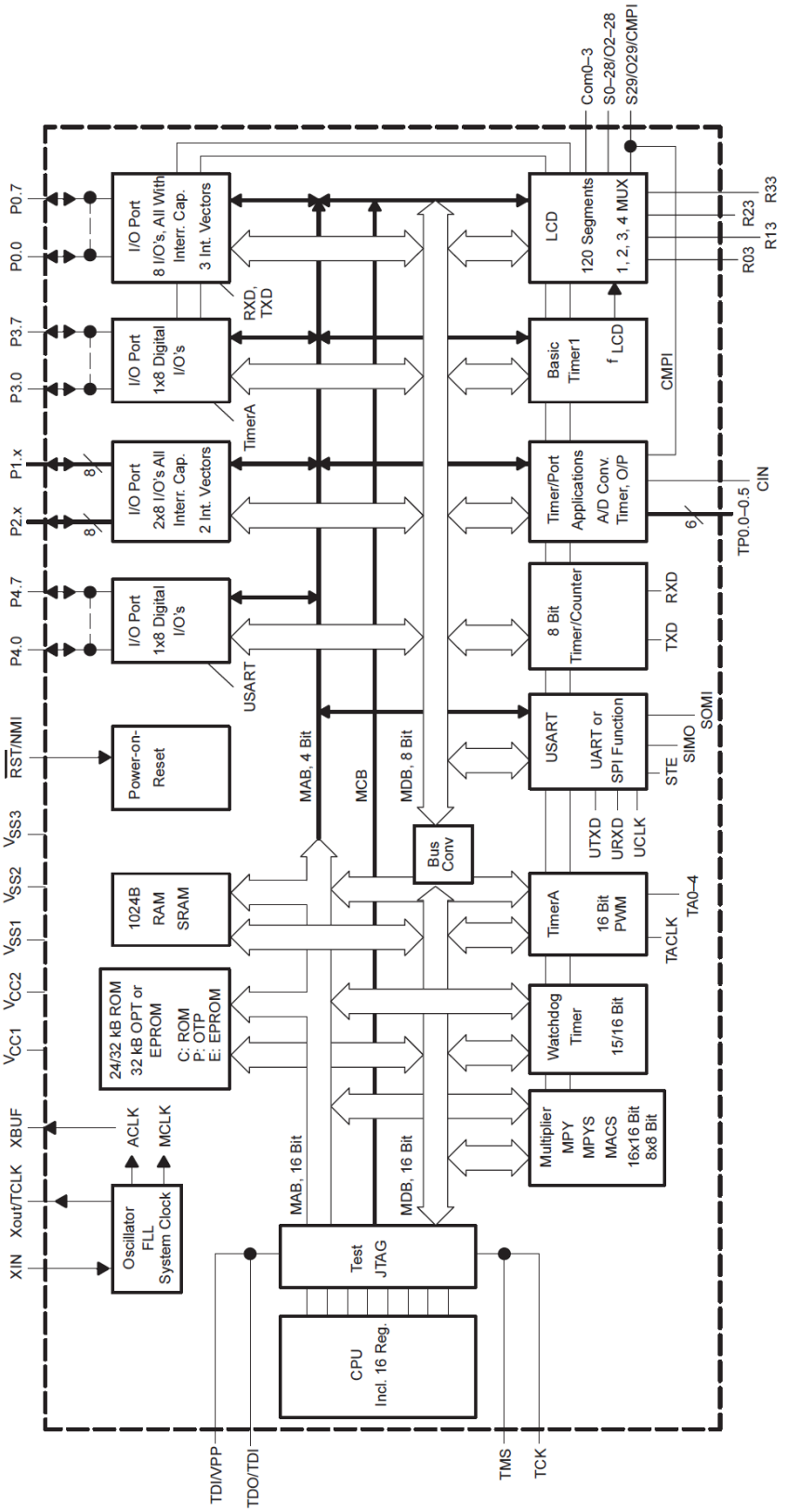


Figure 4.7: MSP430C337 Block Diagram [38]

The Timer/Port module implemented in this chip, has two 8-bit timer/counters. These two can be cascaded to form a 16-bit counter. In addition, an analog to digital conversion can be done with this module. External components are needed to be connected to the pins of this module (according to the datasheet) in order to make the conversion possible. [39][38]

A module named Basic Timer1 is implemented in this chip which provides low-frequency control signals. It contains two 8-bit timers that can be cascaded to form one 16-bit timer. This module is provided to support low current applications. [39][38]

The LCD controller module, provides support for LCDs. This module has an internal memory and does not use the main data memory. [39][38]

There is one USART module implemented in this chip, which provides serial communications. It supports synchronous SPI and asynchronous UART communication protocols. This module does not support I^2C protocol. Transmit and receive channels, each use 8-bit registers and are double buffered. Two dedicated interrupts are assigned to this module (one for transmit and one for receive). In SPI mode, the module supports master and slave modes. [39][38]

For programming, testing and emulation purpose, a JTAG[13] module is implemented in this chip. The JTAG interface port is protected with a fuse. Once this fuse is blown, further access to the chip is not possible via JTAG. [39][38]

To summarize, the following is the characteristics of this microcontroller series [39][38]

- Performance up to 4 MIPS
- Maximum clock frequency: 8 MHz
- 400 μA / MIPS in active state
- 2.5 - 5.5 V
- Maximum current consumption: 4 mA
- 40 GPIO pins
- 12-bit ADC (Slope)
- SPI (Master/Slave support)
- 1x 16-bit Timer, 2x 8-bit Timers, 1x Basic Timer
- Watchdog Timer
- LCD Controller
- Hardware multiplier
- JTAG Interface
- No DAC

[13]Joint Test Action Group, is used for testing printed circuit boards (PCB) using boundary scan. Boundary scan is a method to test interconnects in an integrated circuit or a PCB.

- No I^2C
- No USB support

Table 4.6 shows the characteristics of this microcontroller, with regards to the evaluation criteria.

Performance	Power Consumption per MIPS	I^2C	SPI	ADC	DAC
4 MIPS	12 mW / 4 MIPS = 3	-	M/S	14-bit	-

Memory	WDT	BOR	Temperature	Packaging	Dev. Env.
32 kB	Yes	-	-40 to +85 °C	PQFP, GQFP	CCE, CCS, CrossWorks

Table 4.6: MSP430C337 Criteria Fulfillment
(M: Master, S: Slave, PQFP: Plastic Quad Flat Package, GQFP: Glass Quad Flat Package)

4.1.2.4 MSP430xg461x Series

This series of microcontrollers, use the 16-bit MSP430 RISC-based processor. This processor uses the von Neumann architecture. Figure 4.8 illustrates the functional block diagram of MSP430F261x microcontroller series. Main features of this core are [42]:

- Low power consumption
- 16-bit address bus, 16-bit and 8-bit data bus
- Total 16 registers

  R0 to R3: Program Counter, Stack Pointer, Status Register and Constant Generator

  R4 to R15: General-Purpose Registers

- 51 instructions, three formats and 7 addressing modes

These microcontrollers come in 100-pin QFP[14] and 113-pin BGA[15]. They contain 80 GPIO pins (ten 8-bit I/O ports). The pins can be configured as general purpose I/O or dedicated peripheral interface. Each I/O pin is independently programmable as GPIO even when other pins are used as dedicated peripheral. Port P1 and P2 are interrupt capable. The interrupt for individual I/O lines of P1 and P2, can be enabled and configured. Each port has an 8-bit register. The registers for P7/P8 and P9/P10, have been arranged in such a way that they can be addressed at once as one 16-bit port. Peripherals and the processor are connected together through address, data and control buses. [45][42]

The DMA controller provides data movement without processor intervention. The DMA reduces the power consumption as well as increasing the throughput of peripherals. Reduced power consumption of the system is done by allowing the CPU to remain in

[14]Quad Flat Package
[15]Ball Grid Array

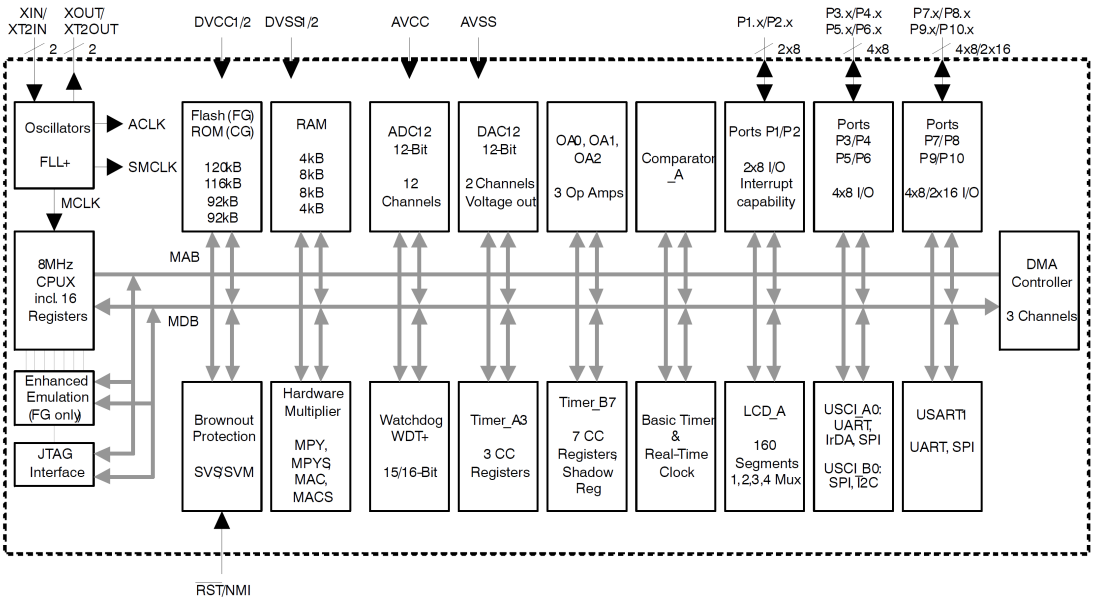


Figure 4.8: MSP430xG461x Block Diagram [42]

sleep mode while DMA moves data from memory to peripheral and vice versa. This module has three independently configurable channels. [45][42]

These microcontrollers have a dedicated multiplier module implemented in them. This module can perform 16×16, 16×8, 8×16 and 8×8 bit operations. After the operands have been loaded in its registers, the result of its operations can be accessed immediately (no additional clock cycles required). [45][42]

The watchdog timer+ (WDT+) module is implemented in these microcontrollers performs a controlled system restart after a software problem occurs. This timer is clocked by an auxiliary 32.768 kHz oscillator. If this timer is not required in an application, it can be configured as an interval timer. Its counter is a 16-bit up counter and is only accessible by the watchdog timer+ control register. The watchdog timer+ module, provides a fail-safe clocking feature. It prevents disabling the WDT+ clock while this module is acting as a watchdog. If the auxiliary or secondary clock sourcing the watchdog timer+ fails, it automatically switches to the master clock. In case the master clock is sourced from a crystal and the crystal has failed, the fail-safe will activate the DCO[16] and will use it as the master clock source. [45][42]

There are two 16-bit timers implemented in these microcontrollers: Timer A3 and Timer B7. Timer A3 is a timer/counter with 3 capture/compare registers. Timer B7 is a timer/counter with 7 capture/compare registers. They both support PWM output and interval timing. Timer interrupts can be generated from the counter on overflow condition and from each capture/compare register. [45][42]

In addition to the two 16-bit timers, a Basic Timer module is implemented. This timer has two independent 8-bit timers. These timers can be cascaded to form one 16-bit

[16]Digitally-Controlled Oscillator

timer/counter. This module is extended and provides a real-time clock (RTC). [45][42]

The LCD controller module (named LCD_A), directly drives LCD displays. It supports static, 2-mux, 3-mux and 4-mux LCDs. It has a display memory implemented in it and does not use the system data memory. The LCD control signal is generated from the Basic Timer module. [45][42]

There are two USCI[17] modules implemented in these chips. They provide support for serial communications. USCI_A0 module supports UART, IrDA and SPI. The USCI_B0 module supports SPI and I^2C protocols. [45][42]

In the UART mode, there are two modes (low-frequency and oversampling) for baud-rate generation. In low-frequency mode, baud-rates can be generated from low frequency clock sources (e.g. 9600 baud (bits per second) from from a 32.768 kHz crystal). In the oversampling mode, higher input clock frequencies are supported. The maximum supported baud-rate, is 460800 baud. [45][42]

The SPI mode on both USCI modules, supports both master and slave modes. Independent 8-bit transmit and receive shift registers and buffer registers have been implemented in this module. Data transmit and receive can be configured as either LSB-first (Least Significant Bit) or MSB-first (Most Significant Bit) transmission. Individual interrupts are generated when module is ready to transmit, or when it has received data. [45][42]

The following are the main features of the I^2C module [45][42]:

- Byte/word format transfer
- 7-bit and 10-bit device addressing modes
- Built in FIFO for read and write buffer
- 16-bit wide data access
- Programmable clock generation
- Start bit detection (slave mode) for auto-wake up from low power mode
- Multi-master and slave mode operation
- Standard mode up to 100 kbps, fast mode up to 400 kbps

In addition to the USCI module, a USART peripheral interface (USART1) is also implemented. This module provides support for UART and SPI modes with one hardware module. [45][42]

The ADC12 module, is a 12-bit SAR[18] analog to digital converter. It has 12 input channels, all individually configurable. There are 16 conversion-result storage registers. It is possible to select either internal or external voltage reference by software. The on-chip (internal) reference voltage generation is also software selectable (1.5 V or 2.5 V). [45][42]

[17] Universal Serial Communication Interface
[18] Successive Approximate Register

The DAC12 module, is a 12-bit voltage output digital to analog converter. This module has 2 output channels. The output voltage resolution can be configured in 8-bit or 12-bit mode. Voltage reference can be selected from internal reference or external. [45][42]

Three configurable low-current general-purpose operation amplifiers (OA0, OA1 and OA2) have been implemented in these microcontrollers. Each OA module can be combined with other modules to create differential amplifiers. Input and output terminal of each OA is software-selectable. Their output signals can be routed to ADC12 module. [45][42]

The comparator module (Comparator_A), is an analog voltage comparator. It supports supply voltage supervision, monitoring external analog signals and precision slope analog to digital conversions. One of the examples of using this comparator is for temperature readings. This comparator has an input short switch bit that can be set to build a simple sample-and-hold comparator. [45][42]

Brown-out reset (BOR) is implemented in these chips. POR signal is generated when either powering up the chip, restarting the chip (from reset pin) or when there is an SVS[19] low condition. BOR circuit detects low supply voltages and resets the chip by triggering the POR signal. The supply voltage supervisor module, monitors the supply voltage. It can be configured to provide a POR or set a flag. SVS is disabled after a brownout reset occurs to conserve current consumption. [45][42]

JTAG[20] module and Bootstrap loader (BSL) are implemented in these microcontrollers. The Flash memory can be programmed using either one of these modules. The JTAG interface port is protected with a fuse. Once this fuse is blown, it is completely disabled and further access to the chip is not possible via JTAG. The Bootstrap loader enables reading and programming the Flash memory or RAM using a UART interface. A 256-bit, user defined password, protects access to the Flash memory via BSL. [45][42]

In addition to the above programming methods, the CPU of these microcontrollers, is able to write to its Flash memory. In this method, the user (programmer) can provide data through any available means (SPI, UART, etc.). The developed software can receive the data and program the Flash memory. This is a completely customizable programming solution and can be used for updating the program memory. [45][42]

To summarize, the following is the characteristics of this microcontroller series [45][42]:

- Performance up to 16 MIPS
- Maximum clock frequency: 16 MHz
- 280 μA / MIPS in active state
- 1.8 - 3.6 V
- Maximum current consumption: 5.920 mA

[19] Supply Voltage Supervisor

[20] Joint Test Action Group, is used for testing printed circuit boards (PCB) using boundary scan. Boundary scan is a method to test interconnects in an integrated circuit or a PCB.

- 80 GPIO pins
- 12-bit ADC Successive Approximative Register (SAR)
- 12-bit DAC
- SPI (Master/Slave support)
- I^2C (Multi-Master and Slave mode support)
- Real-Timer Clock (RTC)
- 2x 16-bit Timers, 2x 8-bit timers
- Watchdog Timer
- Analog Comparator
- Brownout Reset
- Hardware multiplier
- JTAG Interface
- LCD Controller
- No USB support

Table 4.7 shows the characteristics of this microcontroller, with regards to the evaluation criteria.

Performance	**Power Consumption per MIPS**	**I^2C**	**SPI**	**ADC**	**DAC**
16 MIPS	13.024 mW / 16 MIPS = 0.814	MM/S	M/S	12-bit	12-bit

Memory	**WDT**	**BOR**	**Temperature**	**Packaging**	**Dev. Env.**
120 kB	Yes	Yes	-40 to +85 °C	LQFP	CCE, CCS, CrossWorks

Table 4.7: MSP430xg461x Criteria Fulfillment
(M: Master, MM: Multi-Master, S: Slave, LQFP: Low-profile Quad Flat Package)

4.1.3 Atmel AT91SAM ARM Microcontrollers

The AT91SAM series are 32-bit ARM based, Flash microcontrollers, with low power consumption. The AT91SAM7 series have the ARM7TDMI implemented in them as their core. This section covers information about the different Atmel AT91SAM series microcontrollers.

4.1.3.1 AT91M42800A

This microcontroller, uses the 32-bit ARM7TDMI processor core. Figure 4.9 illustrates the functional block diagram of this microcontroller. The architecture of this chip consists of two main buses: Advanced System Bus (ASB) and Advanced Peripheral Bus (APB). The ASB interfaces the processor core with the on-chip memories, the External Bus Interface (EBI) and the AMBA™ Bridge. APB is driven by the AMBA[21] Bridge and interfaces all peripherals with the core. [7]

The EBI implemented in this chip, enables connection to external memories. It is fully programmable and supports 8- or 16-bit devices. It can also emulate a single 16-bit device by using two 8-bit devices. It supports a maximum external address space of 64 MB (mega bytes). This module provides up to 8 chip-selects and a 24-bit address bus (the upper four bits are multiplexed with a chip select). [7]

This microcontroller comes in 144-pin LQFP[22] and BGA[23]. It contains 54 programmable I/O lines (PA0 to PA29 and PB0 to PB23). The Parallel Input/Output Controller (PIO), enables the programmer to select specific pins for dedicated peripheral and general-purpose I/O signal. [7]

There are three system timers embedded in this chip. Real-timer Timer (RTT), Periodic Interval Timer (PIT) and the Watchdog Timer. Typically, their source clock frequency is 32.768 kHz (slow clock). RTT counts elapsed seconds. It has a 20-bit counter implemented in it, which can count up to 1048576 seconds (more than 12 days). PIT provides periodic interrupts. It has a 16-bit down counter implemented in it. Once the counter reaches zero, it is automatically reloaded and restarted. The watchdog timer contains a 16-bit down counter. It allows a maximum watchdog period of 256 seconds, by using the source clock divided by 128. When the watchdog reaches zero, it generates an internal reset and reloads and restarts the counter. It can also generate an interrupt and an external pulse on NWDOVF[24], if the corresponding registers are configured. [7]

There are two Timer/Counter blocks implemented in this chip. Each block contains three identical 16-bit timer/counter channels, which can be used for frequency measurement, interval measurement, event counting, pulse generation (PWM mode) and delay timing. Each module has two multi-purpose input/output lines (TIOA and TIOB), 3 external and 5 internal clock inputs. Each channel, drives an internal interrupt, which can be programmed to generate processor interrupts by using the AIC (Advanced Interrupt Controller). [7]

The Advanced Interrupt Controller (AIC) module, has an 8-level priority, maskable, vectored interrupt controller. It allows the programmer to define a priority between different NIRQ[25] interrupt sources. [7]

The two USART modules implemented in this chip (USART0 and USART1), provide a full-duplex[26] interface. These modules support up to 9-bit data transfers. There is

[21] Advanced Microcontroller Bus Architecture
[22] Low-profile Quad Flat Package
[23] Ball Grid Array
[24] Watchdog Overflow output pin, connected to the Watchdog timer. It is visible on the lower right of Figure 4.9
[25] standard interrupt request
[26] Full-duplex and half-duplex systems, provide communication between two nodes in both directions.

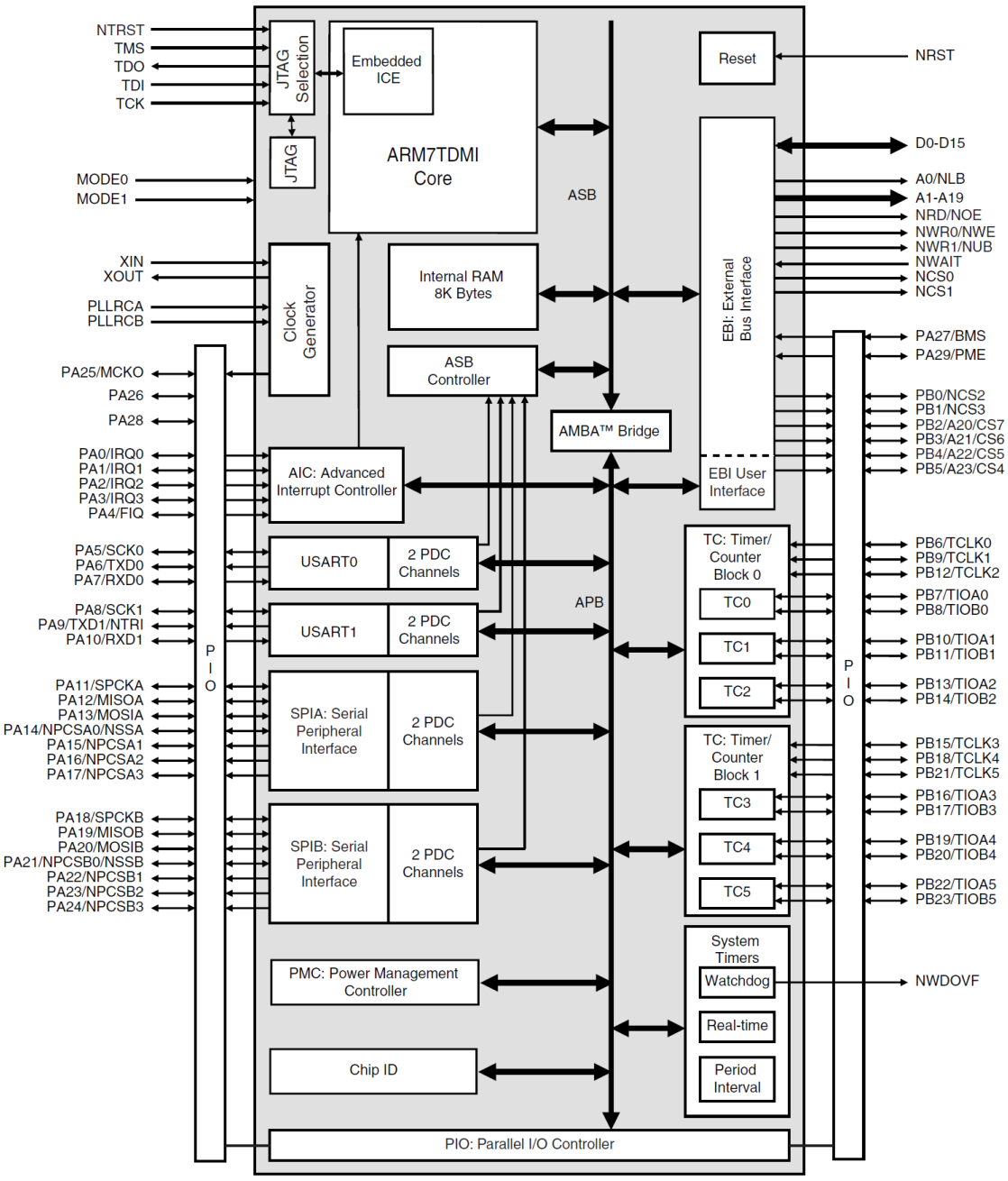


Figure 4.9: AT91M42800A Block Diagram [7]

a baud rate generator implemented in these modules. The clock source for the baud rate generator can either be the external system clock or the internal clock source. In asynchronous mode, baud rate is equal to one-sixteenth of the selected clock source

Full-duplex system allows a simultaneous communication in both directions, while a half-duplex system allows only one direction at a time.

frequency, divided by the non-zero value written in baud rate generator register. In synchronous mode, with internal clock source, the generated baud rate is the selected clock source frequency divided by the value written in the baud rate generator register. In the same mode, but with external clock selected, the baud rate clock is provided directly by the system clock signal. [7]

There are two SPI modules implemented in this chip (SPIA and SPIB). They support both master and slave modes. Each module has four external slave chip-select lines. The data length is programmable to be either 8-bit or 16-bit. Data transmit and receive in this chip is MSB first. Interrupts are generated when data is ready for transmit or data has been received in the receive register. [7]

This chip embeds a 32.768 kHz oscillator inside the Clock Generator module. This module has two PLLs implemented in it. It can generate up to 33 MHz system clock by programming. [7]

The Power Management Controller (PMC) module implemented in this chip, optimizes the power consumption. This is mainly done by controlling the clocking elements (such as system and peripheral clocks, oscillator and PLLs). This module can deactivate the processor and each peripheral individually. A module is deactivated when its clock source is disconnected. [7]

This chip contains two Special Function (SF) registers: Chip Identification and Reset Status. Chip identifier of the AT91M42800A is 0x14280041. This register is read-only. In debug mode (ICE), the ARM core responds with the chip ID to identify the core to the ICE system. The Reset Status register, indicates whether a reset was done by the watchdog timer or an external reset signal (NRST). [7]

Both JTAG[27] and ICE (In-Circuit Emulation) are supported in this chip. The ARM7TDMI standard embedded ICE is supported via the JTAG/ICE port. The ICE provides a complete, easy to use debugging solution. Programming and debugging the the chip is possible via JTAG. [7]

To summarize, the following is the characteristics of this microcontroller [7]:

- Performance up to 30 MIPS
- Maximum clock frequency: 33 MHz
- 280 μA / MIPS in active state
- VDDCORE, VDDIO, VDDPLL: 3.3 V
- Maximum current consumption: -
- 54 GPIO pins
- No ADC or DAC
- SPI (Master/Slave support)
- No I^2C

[27] Joint Test Action Group, is used for testing printed circuit boards (PCB) using boundary scan. Boundary scan is a method to test interconnects in an integrated circuit or a PCB.

- Real-Time Timer (RTT)
- 6x 16-bit Timers (can generate PWM signals)
- Watchdog Timer
- JTAG/ICE Interface
- No LCD Controller
- No USB support
- No Brown-out reset

Table 4.8 shows the characteristics of this microcontroller, with regards to the evaluation criteria.

Performance	**Power Consumption per MIPS**	**I^2C**	**SPI**	**ADC**	**DAC**
30 MIPS	-	-	M/S	-	-

Memory	**WDT**	**BOR**	**Temperature**	**Packaging**	**Dev. Env.**
-	Yes	-	-40 to +85 °C	LQFP, BGA	3rd party programs

Table 4.8: AT91M42800A Criteria Fulfillment
(M: Master, S: Slave, LQFP: Low-profile Quad Flat Package, BGA: Ball Grid Array)

4.1.3.2 AT91M55800A

This microcontroller, uses the 32-bit ARM7TDMI processor core. Figure 4.10 illustrates the functional block diagram of this microcontroller. The chip architecture consists of two main buses: Advanced System Bus (ASB) and Advanced Peripheral Bus (APB). The ASB interfaces the processor core with the on-chip memories, the External Bus Interface (EBI) and the AMBA^{TM} Bridge. APB is driven by the AMBA Bridge and interfaces all peripherals with the core. [9]

The EBI implemented in this chip, enables connection to external memories. It is fully programmable and supports 8- or 16-bit devices. It can also emulate a single 16-bit device by using two 8-bit devices. It supports a maximum external address space of 128 MB (Mega-Bytes). This module has 8 chip-selects and a 24-bit address bus. [9]

This microcontroller comes in 176-pin TQFP[28] and BGA[29]. It contains 58 programmable I/O lines (PA0 to PA29 and PB0 to PB27). The Parallel Input/Output Controller (PIO), enables the programmer to select specific pins for dedicated peripheral and general-purpose I/O signal. [9]

This chip contains a Real-Time Clock (RTC) peripheral. It provides the time-of-day clock and 200 year Gregorian calendar (1900 to 2099). This module has a resolution of one second, with a 32.768 kHz clock source. [9]

[28]Thin Quad Flat Package
[29]Ball Grid Array

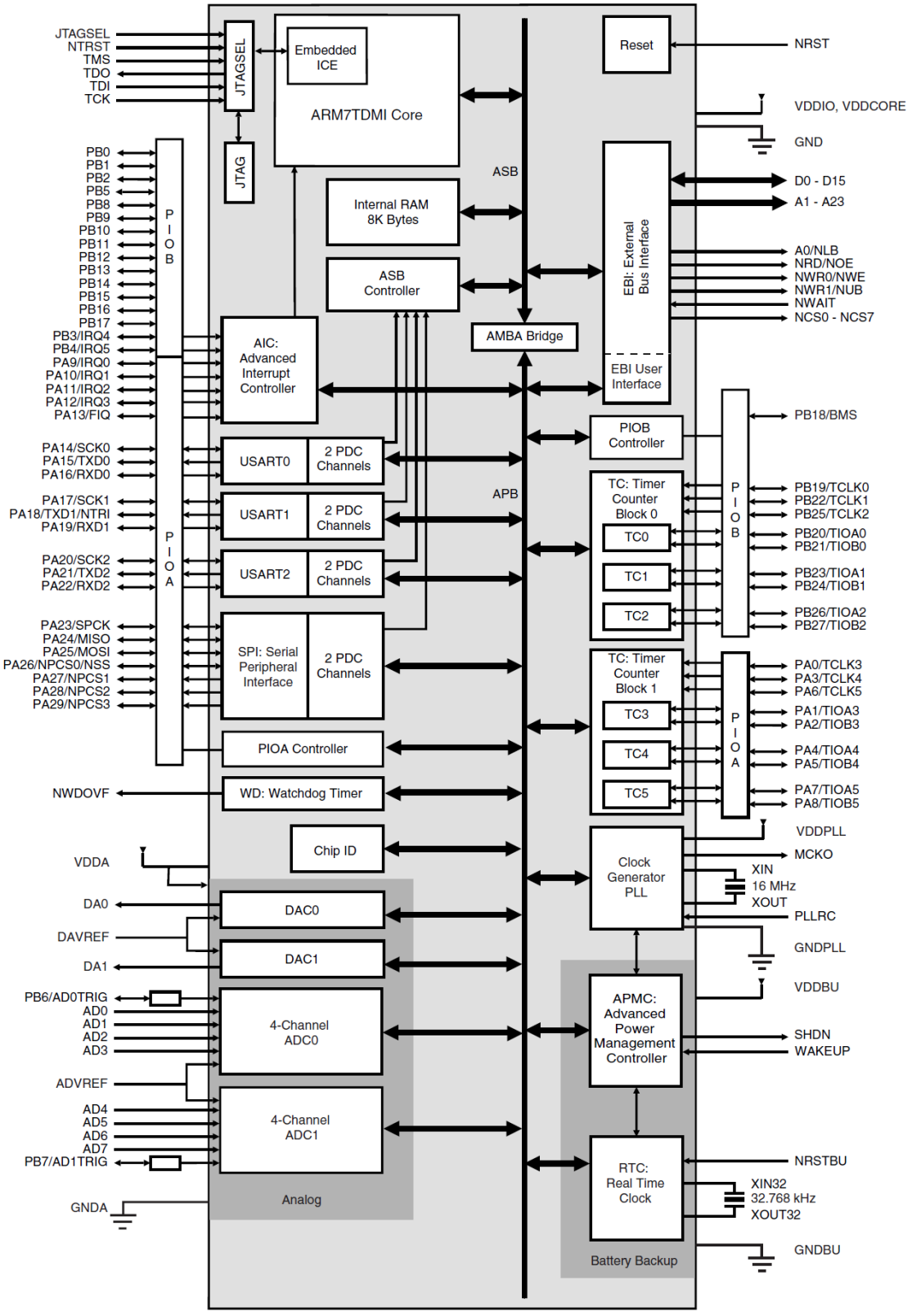


Figure 4.10: AT91M55800A Block Diagram [9]

This chip has an internal watchdog timer to prevent system lock-up. This module has a 16-bit down counter implemented in it. The timeout period is programmable between 4 milliseconds to 8 seconds with a 33 MHz system clock. [9]

There are two Timer/Counter blocks implemented in this chip. Each block contains three identical 16-bit timer/counter channels, which can be used for frequency measurement, interval measurement, event counting, pulse generation (PWM mode) and delay timing. Each module has two multi-purpose input/output lines (TIOA and TIOB), 3 external and 5 internal clock inputs. Each channel, drives an internal interrupt, which can be programmed to generate processor interrupts by using the AIC (Advanced Interrupt Controller). [9]

The Advanced Interrupt Controller (AIC) module, has an 8-level priority, maskable, vectored interrupt controller. It allows the programmer to define a priority between different NIRQ (standard interrupt request) interrupt sources. This module is connected to the NFIQ (Fast Interrupt Request) and NIRQ inputs of the ARM7TDMI processor. NFIQ line can only be asserted from the external FIQ line. The standard IRQ line delivers interrupts generated by peripherals and external interrupt requests (IRQ0 to IRQ5). [9]

The three identical USART modules implemented in this chip (USART0, USART1 and USART2), provide a full-duplex[30] interface. These modules support up to 9-bit (character length) data transfers. There is a baud rate generator implemented in these modules. The clock source for the baud rate generator can either be the external system clock (SCK0, SCK1, SCK2) or the internal clock source. In asynchronous mode, baud rate is equal to one-sixteenth of the selected clock source frequency, divided by the non-zero value written in baud rate generator register. In synchronous mode, with internal clock source, the generated baud rate is the selected clock source frequency divided by the value written in the baud rate generator register. In the same mode, but with external clock selected, the baud rate clock is provided directly by the system clock signal. [9]

Each USART channel (transmit and receive) is connected to one PDC (Peripheral Data Controller). A PDC contains a counter and a pointer. For every bit that is transmitted/received, the counter counts down. Once it reaches zero, it sets the appropriate status registers and generates an interrupt. Data transfer is disabled until a new non-zero value is written in the registers. [9]

There is one SPI module implemented in this chip. It supports both master and slave modes. It has four external slave chip-select lines, which can be connected to up to 15 devices. The data length is programmable from 8-bit to 16-bit. Data transmit and receive in this module is MSB first. Interrupts are generated when data is ready for transmit or data has been received in the receive register. [9]

There are two identical 4-channel 10-bit SAR[31] ADC modules (ADC0 and ADC1). They can operate in 8-bit or 10-bit mode. Both ADCs share the same reference voltage (ADVREF) and power supply pins (VDDA and GNDA). [9]

There are two 10-bit DAC modules implemented in this chip (DAC0 and DAC1).

[30] Full-duplex and half-duplex systems, provide communication between two nodes in both directions. Full-duplex system allows a simultaneous communication in both directions, while a half-duplex system allows only one direction at a time.

[31] Successive Approximate Register

These two modules are identical. They can operate in 8-bit or 10-bit mode. Both DACs share the same reference voltage (DAVREF) and power supply pins (VDDA and GNDA). [9]

The Advanced Power Management Controller (APMC) module implemented in this chip, optimizes the power consumption. This is mainly done by controlling the clocking elements (such as system and peripheral clocks, oscillator and PLLs). This module can deactivate the processor and each peripheral individually. A module is deactivated when its clock source is disconnected. [7]

The implemented Clock Generator module, provides all the clock sources for the chip. It consists of the main oscillator, a PLL and a clock selection logic. The main oscillator is designed for a crystal with frequency between 3 to 20 MHz. The main oscillator can be configured to be bypassed. In this case, any frequency can be the input on XIN pin. This module can be deactivated for reduced power consumption. [9]

This chip contains two Special Function (SF) registers: Chip Identification and Reset Status. Chip identifier of the AT91M55800A is 0x15580040. This register is read-only. In debug mode (ICE), the ARM core responds with the chip ID to identify the core to the ICE system. The Reset Status register, indicates whether a reset was done by the watchdog timer or an external reset signal (NRST). [9]

Both JTAG[32] and ICE (In-Circuit Emulation) are supported in this chip. The ARM7TDMI standard embedded ICE is supported via the JTAG/ICE port. The ICE provides a complete, easy to use debugging solution. Programming and debugging the the chip is possible via JTAG. [9]

To summarize, the following is the characteristics of this microcontroller [9]:

- Maximum performance: -
- Maximum clock frequency: 33 MHz
- mA / MIPS in active state: n/a
- VDDCORE: 3.0V - 3.3V; VDDIO: 3.0V - 5V
- Maximum current consumption: -
- 58 programmable IO lines
- 10-bit ADC
- 10-bit DAC
- SPI (Master/Slave support)
- No I^2C
- Real-Time Clock (RTC)
- 6x 16-bit Timers (can generate PWM signals)

[32]Joint Test Action Group, is used for testing printed circuit boards (PCB) using boundary scan. Boundary scan is a method to test interconnects in an integrated circuit or a PCB.

- Watchdog Timer
- JTAG/ICE Interface
- No LCD Controller
- No USB support
- No Brown-out reset

Table 4.9 shows the characteristics of this microcontroller, with regards to the evaluation criteria.

Performance	**Power Consumption per MIPS**	**I^{2}C**	**SPI**	**ADC**	**DAC**
-	-	-	M/S	10-bit	-

Memory	**WDT**	**BOR**	**Temperature**	**Packaging**	**Dev. Env.**
-	Yes	-	-40 to +85 °C	LQFP, BGA	3rd party programs

Table 4.9: AT91M55800A Criteria Fulfillment
(M: Master, S: Slave, LQFP: Low-profile Quad Flat Package, BGA: Ball Grid Array)

4.1.3.3 AT91RM9200

This microcontroller, uses the 32-bit ARM920T processor core. Figure 4.11 illustrates the functional block diagram of this microcontroller. The processor is connected to the Memory Controller module. This module manages the ASB (Advanced System Bus). APB (Advanced Peripheral Bus) is connected to this module via the Peripheral DMA (Dynamic Memory Access) Controller and Peripheral Bridge. APB interfaces the processor with all peripherals. [13]

The Peripheral DMA Controller (PDC) module, transferes data between interfaces all on-chip serial peripherals with on-chip and external memories. This is done without processor intervention. Memory space of each peripheral has the user interface of PDC channel integrated in it. The channels are implemented in pair (total 20 channels). One channel in pair is dedicated to the receiving channel and the other is dedicated to the transmitting channel of each peripheral (USART, UART, SPI and SSC). [13]

The EBI (External Bus interface) implemented in this chip, enables data transfer between several external devices and the Memory Controller module. It features static memory, Burst Flash and SDRAM[33] memory controllers. These controllers support SRAM, PROM, EPROM, EEPROM, Flash, SDRAM, Burst Flash, CompactFlash and the NAND Flash/SmartMedia. EBI supports data transfer to up to 8 external devices. Data transfer is done through a 16-bit or 32-bit data bus and an address bus of up to 26-bits (up to 64 MB (Mega Bytes) addressable). [13]

This microcontroller comes in 208-pin PQFP[34] and 256-pin BGA[35]. There are up to four PIO (Parallel I/O) controllers in this chip. PIOA and PIOB, multiplex peripheral

[33]Synchronous Dynamic Random Access Memory
[34]Plastic Quad Flat Package
[35]Ball Grid Array

I/O lines. PIOC, multiplexes the data bus bits from 16 to 31 and several EBI control signals. PIOD (only available on the 256-pin BGA package) multiplexes output peripheral set and ETM port. Each PIO controller can control up to 32 lines. In total there are up to 122 Programmable I/O lines. [13]

The Real-Time Clock (RTC) peripheral implemented on this chip provides the time-of-day clock and 200 year Gregorian calendar. This module has a resolution of one second, with a 32.768 kHz clock source. [13]

This chip contains a System Timer, where three different free-running timers are integrated in it: a 16-bit Period Interval Timer (PIT), a 16-bit Watchdog Timer (WDT) and a 20-bit Real-Time Timer (RTT). [13]

The 16-bit Watchdog Timer runs on 32.768 kHz clock frequency (slow clock). It uses this clock frequency divided by 128, to provide the maximum watchdog period of up to 256 seconds. [13]

The Real-Time Timer is built around a 20-bit counter and is used to count the elapsed seconds. It uses the slow clock (32.768 kHz) divided by a programmable value and can count up to 1048576 seconds (more than 12 days). [13]

The Periodic Interval Timer, provides periodic interrupts. It is built around a 16-bit down counter and is fed by the slow clock. Once the counter reaches zero, it is automatically reloaded and restarted. [13]

The Advanced Interrupt Controller (AIC) module, has an 8-level priority, maskable, vectored interrupt controller. It allows the programmer to define a priority between different nIRQ (standard interrupt request) interrupt sources. This module drives the nFIQ (Fast Interrupt Request) and nIRQ inputs of the ARM processor. nFIQ line can only be asserted from the external FIQ line. The standard IRQ line delivers interrupts generated by peripherals and external interrupt requests (IRQ0 to IRQ6). [13]

The Power Management Controller (PMC) module implemented in this chip, generates all the clocks. It is integrated with the main oscillator, slow clock (32.768 kHz) oscillator and two PLLs. It provides clock for the processor (PCK), USB Host Port, USB Device Port and other peripherals. This module has four modes of operation (normal, idle, slow clock and standby). It can lower the power consumption (idle, slow clock and standby modes) by altering or disabling the processor or peripheral clocks. [13]

There are two Timer/Counter blocks implemented in this chip. Each block contains three identical 16-bit timer/counter channels, which can be used for frequency measurement, interval measurement, event counting, pulse generation (PWM mode) and delay timing. Each channel is user-configurable and has two multi-purpose input/output lines, 3 external and 5 internal clock inputs. Each channel, drives an internal interrupt, which can be programmed to generate processor interrupts by using the AIC (Advanced Interrupt Controller). [13]

Synchronous Serial Controller (SSC), is used to provide a synchronous communication with external devices. There are three SSC modules implemented in this chip. Each module has independent transmitter and receiver with a common clock divider. It supports many serial synchronous communication protocols (such as Long Frame Sync, Short Frame Sync, I^2S). Each module is connected to two PDC channels, which reduces processor intervention. Digital to analog conversion (DAC) is possible through these modules. [13]

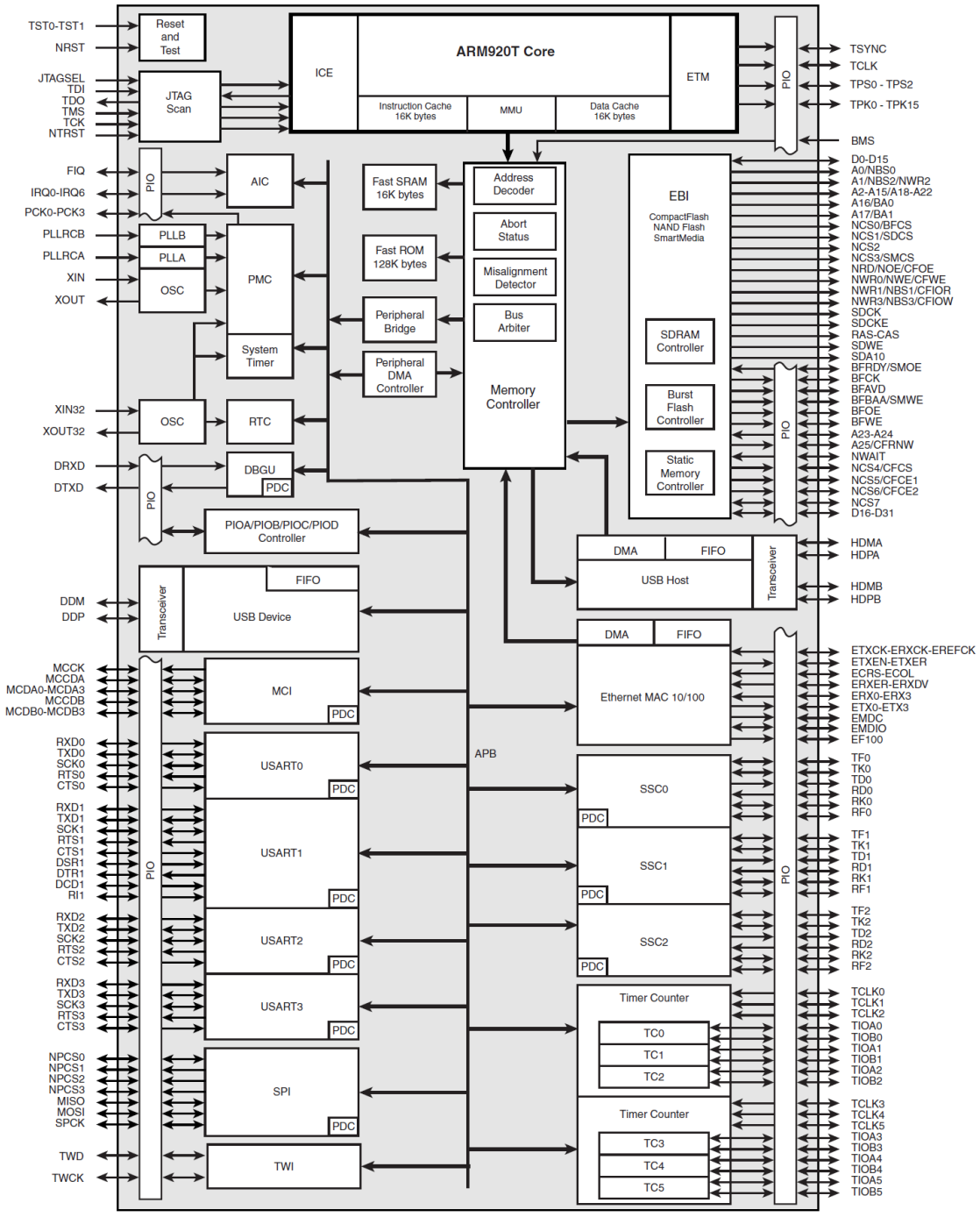


Figure 4.11: AT91RM9200 Block Diagram [13]

The Ethernet MAC (EMAC) module implemented in this chip, features 10 and 100 Mbits/sec data throughput. It has a 28-byte transmit and a 28-byte receive FIFOs. This module can transfer data in two modes: Media-independent Interface (MII) and

Reduced Media-independent Interface (RMII)[36]. EMAC features full- and half-duplex[37] operation and a DMA interface. [13]

This chip contains USB 2.0 Full Speed (12 Mbit/sec) host and device ports. The USB host module, has dual on-chip transceivers, an integrated FIFO and dedicated DMA channels. The USB device module, contains an on-chip transceiver and 2 KB (kilo bytes) of configurable FIFO. The USB device port complies with USB v2.0 Full Speed specification. The USB host port, complies with both USB v2.0 full speed and low speed specifications. [13]

The Multimedia Card Interface (MCI) provides support for MMC (MultiMedia Card) Specification v2.2 as well as SD Memory Card Specification v1.0. This interface includes command, data, timeout and response registers along with an error detection logic. It automatically handles command transmission and when required, reception of data and responses, with limited processor overhead. This module supports interfacing of one slot, which can be used to interface with a MMC (MultiMedia Card) bus (maximum 30 cards) or with a SD Memory Card. The MMC communication is based on a 7-pin (clock, command, 1 data, 3 power lines and 1 received) interface and the SD Memory Card on a 9-pin (clock, command, 3 power and 4 data lines) interface. The SD memory Card interface, supports MMC operations. Their main difference (SDMC and MMC) is in their bus topology and initialization process. [13]

The four USART modules implemented in this chip (USART0, USART1, USART2 and USART3), provide a full-duplex synchronous or asynchronous serial communication interface. These modules support MSB- or LSB-first, 5- to 9-bit (character length) communication. They feature a programmable Baud Rate Generator. The clock source for the baud rate generator can either be the external system clock (SCK0, SCK1, SCK2, SCK3) or the internal clock source. The maximum baud rate of 200000 baud (bit/sec), can be generated. Each USART channel (transmit and receive) is connected to one PDC (Peripheral Data Controller). All USART modules support IrDA modulation and demodulation. USART1 supports Modem Mode communication. USART0, USART2 and USART3, do not have all modem signals implemented. [13]

There is one SPI module implemented in this chip. It supports both master and slave modes. It also provides communication between processors, in case an external processor is connected. It has four external slave chip-select lines, which provides communication with up to 15 devices. The data length is programmable from 8-bit to 16-bit. Data transmit and receive in this module is MSB first. Each channel (receive and transmit) is connected to PDC. SPI interrupt line is connected to AIC (Advanced Interrupt Controller) and can be configured to be generated when data is ready for transmit or data has been received in the receive register. [13]

The Two-Wire Interface (TWI/I^2C) implemented in this chip, is made of one data line and one clock line, with up to 400 kbits/sec speed. This module only supports master mode with sequential or single-byte access. [13]

[36] RMII aims at reducing the pin count

[37] Full-duplex and half-duplex systems, provide communication between two nodes in both directions. Full-duplex system allows a simultaneous communication in both directions, while a half-duplex system allows only one direction at a time.

This chip includes JTAG[38], ICE (In-Circuit Emulation) and a dedicated UART debug channel (DBGU). It also has an embedded real time trace which provides debugging capability for real-time applications. The ARM9TDMI embedded ICE is supported via the JTAG/ICE port. The ICE provides a complete, easy to use debugging solution. Programming and debugging the the chip is possible via JTAG. The Debug Unit (DBGU) provides a UART communication that can be used for debugging purposes. This module can be used to upload a program into the internal SRAM, which can be activated by the Boot Program (integrated bootloader and boot uploader). [13]

To summarize, the following is the characteristics of this microcontroller [13]:

- Performance up to 200 MIPS
- Maximum clock frequency: 180 MHz
- mA / MIPS in active state: 0.122
- VDDCORE, VDDPLL, VDDOSC: 1.8V; VDDIO: 3.0V - 3.6V
- Maximum current consumption: 24.4 mA @ 180 MHz
- 122 programmable IO lines
- No ADC
- DAC via SSC (Synchronous Serial Controller) modules
- SPI (Master/Slave support)
- I^2C(Master mode support)
- Real-Time Clock (RTC)
- Real-Time Timer (RTT)
- 6x 16-bit Timers/Counters (can generate PWM signals)
- Synchronous Serial Controller
- Multimedia Card Interface (MCI)
- Watchdog Timer
- USB Host Controller
- USB Device Controller
- Ethernet MAC 10/100 Controller
- DMA Controller

[38]Joint Test Action Group, is used for testing printed circuit boards (PCB) using boundary scan. Boundary scan is a method to test interconnects in an integrated circuit or a PCB.

- JTAG/ICE/DBGU Interface
- No LCD Controller
- No Brown-out reset

Table 4.10 shows the characteristics of this microcontroller, with regards to the evaluation criteria.

Performance	**Power Consumption per MIPS**	**I^{2}C**	**SPI**	**ADC**	**DAC**
200 MIPS	43.92 mW / 200 MIPS = 0.2196	M	M/S	-	-

Memory	**WDT**	**BOR**	**Temperature**	**Packaging**	**Dev. Env.**
128 kB	Yes	-	-40 to +85 °C	PQFP, BGA	3rd party programs

Table 4.10: AT91RM9200 Criteria Fulfillment
(M: Master, S: Slave, PQFP: Plastic Quad Flat Package, BGA: Ball Grid Array)

4.1.3.4 AT91SAM7A3

This microcontroller, uses the 32-bit ARM7TDMI processor core. Figure 4.12 illustrates the functional block diagram of this microcontroller. The processor is connected to the Memory Controller module. This module manages the ASB (Advanced System Bus). APB (Advanced Peripheral Bus) is connected to this module via the Peripheral Data Controller and Peripheral Bridge. APB interfaces the processor with all peripherals. [8]

The Peripheral DMA Controller (PDC) module[39], transfers data between all on-chip serial peripheral interfaces with on-chip and external memories. This data transfer is done without processor intervention. Memory space of each peripheral has the user interface of PDC channel integrated in it. There are total number of 19 channels implemented. Several peripherals use channels in pair. In a pair, one channel is dedicated to the receiving channel and the other is dedicated to transmitting channel of each peripheral (USART, DBGU, SPI and SSC). MCI and ADC each use only one channel. [8]

This microcontroller comes in 100-pin LQFP[40]. There are two Parallel Input/Output controllers implemented in this chip (PIOA and PIOB). In total, there are 62 programmable I/O lines multiplex with up to two peripheral I/O lines. [8]

This chip contains three system timers: a Real-Time Timer (RTT), a Periodic Interval Timer (PIT) and a Watchdog Timer (WDT). The Real-Time Timer is built around a 32-bit counter and is used to count the elapsed seconds. It uses the slow clock (32.768 kHz) divided by a programmable 16-bit value and can count up to 2^{32} seconds (more than 136 years). [8]

[39]Atmel AT91SAM7A3 datasheet, has used "Peripheral Data Controller" and Peripheral DMA Controller, while pointing at the PDC. In this report, the name Peripheral DMA Controller is used.

[40]Low-profile Quad Flat Package

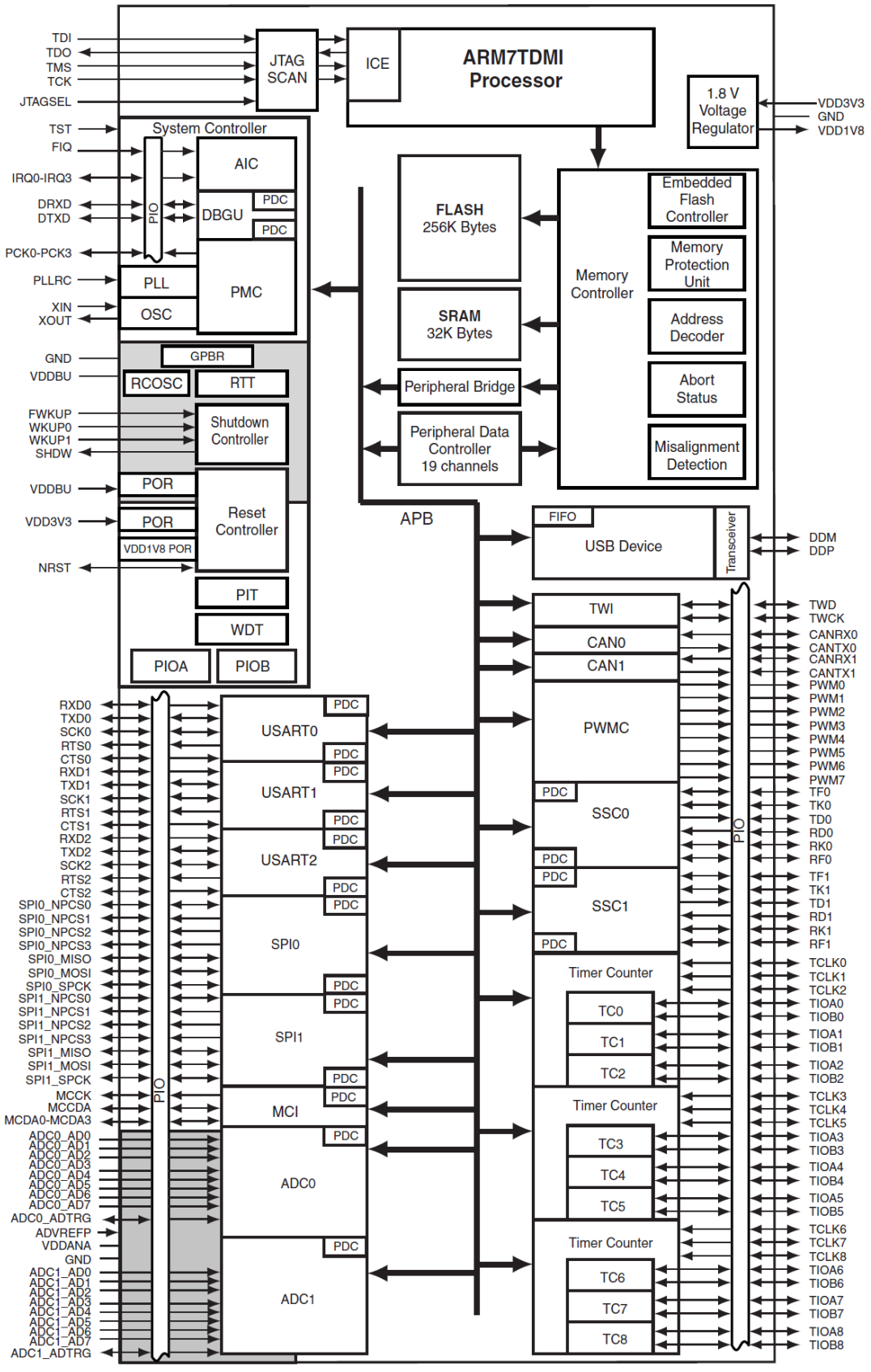


Figure 4.12: AT91SAM7A3 Block Diagram [8]

The Periodic Interval Timer, provides periodic interrupts. It is built around a 20-bit CPIV[41] counter and a 12-bit PICNT[42] counter. Both counters work at the Master Clock (main peripheral clock) divided by 16. Once the counter reaches zero, it is automatically reloaded and restarted. [8]

The 12-bit Watchdog Timer runs on 32.768 kHz clock frequency (slow clock) and can count up to 16 seconds. It is able to generate a general reset or only a processor reset. Independent of the type of reset, Error and Underflow status bits, trigger an interrupt. The value in watchdog timer register can only be changed with a processor reset. [8]

The Advanced Interrupt Controller (AIC) module, has an 8-level priority, maskable, vectored interrupt controller. It allows the programmer to define a priority between different nIRQ (standard interrupt request) interrupt sources. This module drives the nFIQ (Fast Interrupt Request) and nIRQ inputs of the ARM processor. nFIQ line can only be asserted from the external FIQ line. The standard IRQ line delivers interrupts generated by peripherals and external interrupt requests (IRQ0 to IRQ3). [8]

The Power Management Controller (PMC) module implemented in this chip, generates all system and peripheral clocks. It also optimizes the power consumption by controlling the clocks. It is integrated with the main oscillator, slow clock (32.768 kHz) oscillator and the PLL. It provides clock for the processor (PCK), USB Device Port and other peripherals. [8]

This chip features a Shutdown Controller module. It controls and manages the main power supplies (3.3 V and 1.8 V) along with wake-up detection. A Force Wake-up (FWKUP) input signal is dedicated. The wake-up signals (WKUP0, WKUP1, FWKUP) can be connected to any signal to wake the system up. This module is continuously clocked by Slow Clock (32.768 kHz). The Power Management Controller does not control this module. The SHDW[43] pin is controlled by software. [8]

A Reset Controller module is implemented in this chip. It is based on power-on reset (POR) cells and handles all the system resets. This includes reporting which reset last occurred. This module handles the following resets (descending order): backup reset, wake-up reset, watchdog reset, software reset and user reset. [8]

There are three Timer/Counter blocks implemented in this chip. Each block contains three identical 16-bit timer/counter channels, which can be used for frequency measurement, interval measurement, event counting, pulse generation (PWM mode) and delay timing. Each channel is user-configurable and has two multi-purpose input/output lines, 3 external and 5 internal clock inputs. Each channel, drives an internal interrupt, which can be programmed to generate processor interrupts by using the AIC (Advanced Interrupt Controller). [8]

Synchronous Serial Controller (SSC), is used to provide a synchronous communication with external devices. There are three SSC modules implemented in this chip. Each module has independent transmitter and receiver with a common clock divider. It supports many serial synchronous communication protocols (such as Long Frame Sync, Short Frame Sync, I^2S). Each module is connected to two PDC channels, to reduce processor intervention. Digital to analog conversion (DAC) is possible through these

[41] Current Periodic Interval Value

[42] Periodic Interval Counter

[43] Shutdown output pin, connected directly to the Shutdown Controller

modules. [8]

The PWMC (Pulse Width Modulation Controller) module contains 8 channels. Each channel uses one of the clocks (provided from the internal clock generator) and can control one square output waveform. All channels are double buffered. This prevents the generation of an unexpected output waveform, while modifying the either period or duty-cycle. [8]

There are two CAN (Controller Area Network) modules implemented in this chip. Each module provides all features necessary for implementation of CAN serial communication protocol. It has the ability to handle all frame types (Data, Remote, Error and Overload) with a bitrate of up to 1 Mbit/sec. [8]

The Two-wire Interface (TWI/I^2C) implemented in this chip, is made of one data line and one clock line, with up to 400 kbits/sec speed. This module only supports master mode with sequential or single-byte access. [8]

This chip contains USB 2.0 Full Speed (12 Mbit/sec) device ports. it integrates an on-chip transceiver and 2376-byte of configurable FIFO. Each endpoint is configurable in one of several USB transfer types[44]. One or two dual-port RAM (DPR) banks can be associated with this module. If two banks are associated, one will be handled with the USB Device module and the other will be handled by the processor. In isochronus endpoints, this is mandatory so the device can maintain the maximum bandwidth (1MB/s) while working with two DPR banks. [8]

The Multimedia Card Interface (MCI) provides support for MMC (MultiMedia Card) Specification v2.2 as well as SD Memory Card Specification v1.0. This interface includes command, data, timeout and response registers along with an error detection logic. It automatically handles command transmission and when required, reception of data and responses, with limited processor overhead. This module supports interfacing of one slot, which can be used to interface with a MMC (MultiMedia Card) bus (maximum 30 cards) or with a SD Memory Card. The MMC communication is based on a 7-pin (clock, command, 1 data, 3 power lines and 1 received) interface and the SD Memory Card on a 9-pin (clock, command, 3 power and 4 data lines) interface. The SD memory Card interface, supports MMC operations. Their main difference (SDMC and MMC) is in their bus topology and initialization process. [8]

The three USART modules implemented in this chip (USART0, USART1 and USART2), provide a full-duplex[45] synchronous or asynchronous serial communication interface. These modules support MSB- or LSB-first, 5- to 9-bit (character length) communication. They feature a programmable Baud Rate Generator. The clock source for the baud rate generator can either be the external system clock (SCK0, SCK1, SCK2) or the internal clock source. The maximum baud rate of 115200 baud (bit/sec), can be generated. Each USART channel (transmit and receive) is connected to one PDC (two for each module). [8]

There are two SPI modules implemented in this chip that support both master and

[44]The USB specification defines 4 transfer types: Control Transfer, Interrupt Transfer, Isochronus Transfer, Bulk Transfer [55]

[45]Full-duplex and half-duplex systems, provide communication between two nodes in both directions. Full-duplex system allows a simultaneous communication in both directions, while a half-duplex system allows only one direction at a time.

slave modes. Each module provides communication between processors, in case an external processor is connected. It has four external slave chip-select lines, which provide communication with up to 15 devices. The data length is programmable from 8-bit to 16-bit. Data transmit and receive in this module is MSB first. Each channel (receive and transmit) of each module is connected to PDC. SPI interrupt line is connected to AIC (Advanced Interrupt Controller) and can be configured to be generated when data is ready for transmit or data has been received in the receive register. [8]

There are two 8-channel 10-bit SAR (Successive Approximate Register) ADC (Analog to Digital Converter) modules implemented in this chip. Each support 8- or 10-bit resolution mode. The conversion result is stored in a common register (for all channels) and in a channel-dedicated register. It performs the conversion in a full range between zero volts and the reference voltage pin value. Both ADCs share the same reference voltage. [8]

This chip includes JTAG (Joint Test Action Group), ICE (In-Circuit Emulation) and a dedicated UART debug channel (DBGU). It also has an embedded real time trace which provides debugging capability for real-time applications. The ARM9TDMI embedded ICE is supported via the JTAG/ICE port. The ICE provides a complete, easy to use debugging solution. Programming and debugging the chip is possible via JTAG. The Debug Unit (DBGU) provides a UART communication that can be used for debugging purposes. This module can be used to upload a program into the internal SRAM, which can be activated by the Boot Program (integrated bootloader and boot uploader). [8]

To summarize, the following is the characteristics of this microcontroller [8]:

- Performance up to 54 MIPS (0.9 MIPS/MHz)
- Maximum clock frequency: 60 MHz
- mA / MIPS in active state: 1.3
- 3.3 V Single supply
- Maximum current consumption: 70 mA @ 60 MHz
- 62 programmable IO lines
- 2x 10-bit SAR ADC
- DAC via SSC (Synchronous Serial Controller) modules
- 2x SPI (Master/Slave support)
- I^2C(Master mode support)
- Real-Time Timer (RTT)
- 9x 16-bit Timers/Counters (can generate PWM signals)
- 16-bit PWM Controller
- 2x Synchronous Serial Controller

- Multimedia Card Interface (MCI)
- Watchdog Timer
- USB Device Controller
- DMA Controller
- On-Chip RC Oscillator
- JTAG/ICE/DBGU Interface
- No LCD Controller
- No Brown-out reset

Table 4.11 shows the characteristics of this microcontroller, with regards to the evaluation criteria.

Performance	**Power Consumption per MIPS**	**I^{2}C**	**SPI**	**ADC**	**DAC**
54 MIPS	231 mW / 54 MIPS = 4.28	M	M/S	10-bit	-

Memory	**WDT**	**BOR**	**Temperature**	**Packaging**	**Dev. Env.**
256 kB	Yes	-	-40 to +85 °C	LQFP	3rd party programs

Table 4.11: AT91SAM7A3 Criteria Fulfillment
(M: Master, S: Slave, LQFP: Low-profile Quad Flat Package)

4.1.3.5 AT91SAM7L128

This microcontroller, uses the 32-bit ARM7TDMI processor core. Figure 4.13 illustrates the functional block diagram of this microcontroller. The processor is connected to the Memory Controller module. This module manages the ASB (Advanced System Bus). APB (Advanced Peripheral Bus) is connected to this module via the Peripheral Data Controller and Peripheral Bridge. APB interfaces the processor with all peripherals. [10]

The Peripheral DMA Controller (PDC) module[46], transfers data between all on-chip serial peripheral interfaces with on-chip and external memories. This data transfer is done without processor intervention. Memory space of each peripheral has the user interface of PDC channel integrated in it. There are total number of 11 channels implemented. Most channels are implemented in pair (except for ADC). One channel in pair is dedicated to the receiving channel and the other is dedicated to transmitting channel of each peripheral (USART, DBGU, SPI and TWI). ADC uses only one channel. [10]

This microcontroller comes in 128-pin LQFP[47] and 144-pin BGA[48]. There are three Parallel Input/Output controllers implemented in this chip (PIOA, PIOB and PIOC).

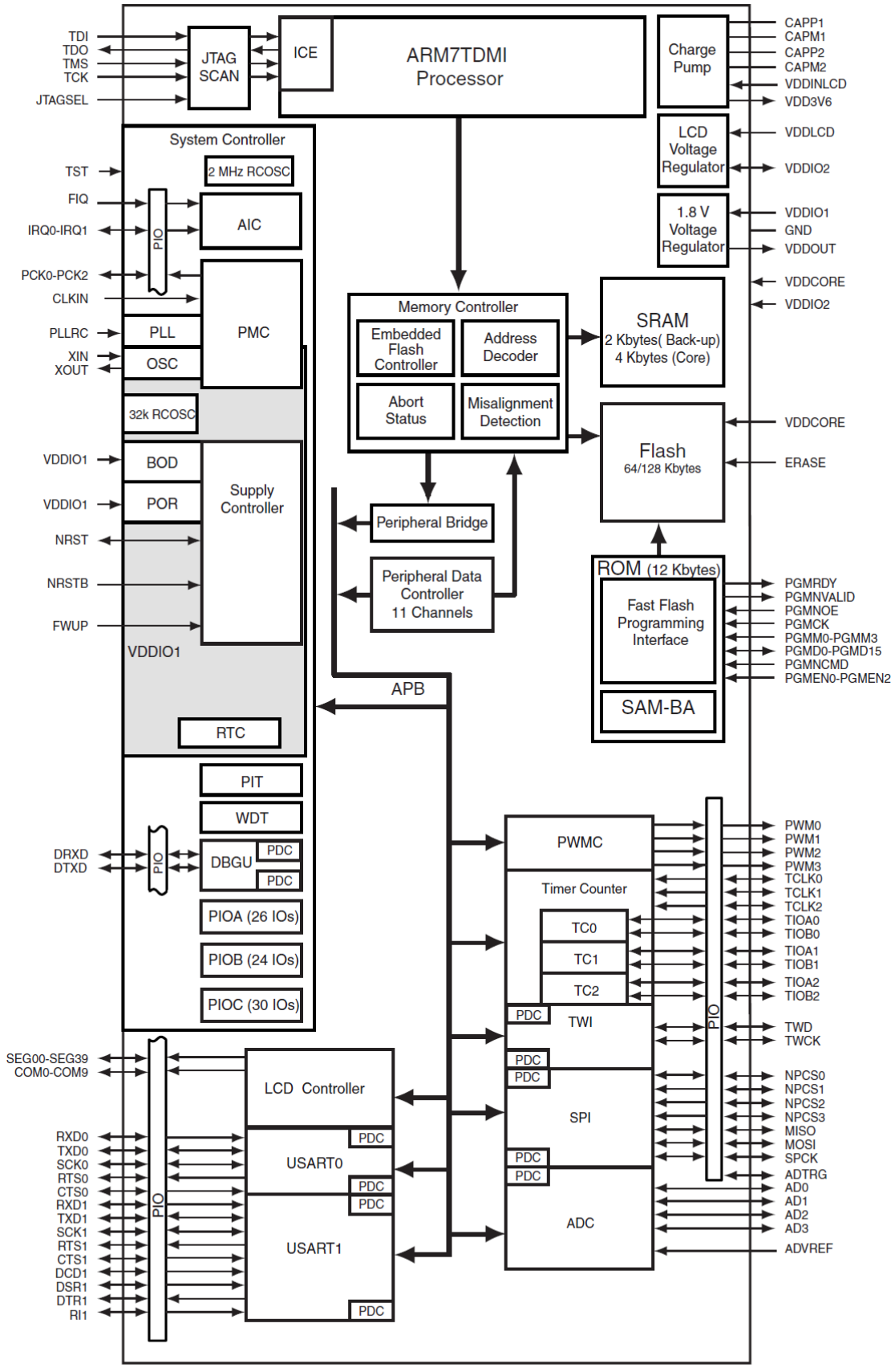


Figure 4.13: AT91SAM7L64/128 Block Diagram [10]

In total there are 80 programmable I/O lines multiplexed with up to two peripheral I/O lines. [10]

The Supply Controller implemented in this chip, controlls the power supply of the processor, Flash memory, backup SRAM, LCD controller and the Real-Time Clock. It is based on the NRSTB[49] pin, the power-on reset (POR) circuit and Brown-out Detector (BOD) circuit. It controls internal resets and saves the last reset status. [10]

The Real-Time Clock (RTC) peripheral implemented on this chip provides the time-of-day clock and 200 year Gregorian calendar (1900 to 2099). This module has a resolution of one second, with a continuous 32.768 kHz clock source. [10]

The Periodic Interval Timer, provides periodic interrupts. It is built around a 20-bit CPIV[50] counter and a 12-bit PICNT[51] counter. Both counters work at the Master Clock (main peripheral clock) divided by 16. Once the counter reaches zero, it is automatically reloaded and restarted. [10]

The 12-bit Watchdog Timer runs on 32.768 kHz clock frequency (slow clock) and can count up to 16 seconds. It is able to generate a general reset or only a processor reset. Independent of the type of reset, Error and Underflow status bits, trigger an interrupt. The value in watchdog timer register can only be changed with a processor reset. [10]

The Advanced Interrupt Controller (AIC) module, has an 8-level priority, maskable, vectored interrupt controller. It allows the programmer to define a priority between different nIRQ (standard interrupt request) interrupt sources. This module drives the nFIQ (Fast Interrupt Request) and nIRQ inputs of the ARM processor. nFIQ line can only be asserted from the external FIQ line. The standard IRQ line delivers interrupts generated by peripherals and external interrupt requests (IRQ0 and IRQ1). [10]

The Power Management Controller (PMC) module implemented in this chip, generates all system and peripheral clocks. It also optimizes the power consumption by controlling the clocks. It is integrated with the main oscillator, slow clock (32.768 kHz) oscillator and the PLL. It provides clock for the processor (PCK) and programmable clock outputs (PCK0 to PCK2 signals). [10]

A Reset Controller module is implemented in this chip. It is based on power-on reset (POR) cells and handles all the system resets. This includes reporting which reset last occurred. This module handles the following resets (descending order): General reset, Backup reset, Watchdog reset, Software reset and User reset. [10]

There is one Timer/Counter block implemented in this chip. It contains three identical 16-bit timer/counter channels, which can be used for frequency measurement, interval measurement, event counting, pulse generation (PWM mode) and delay timing. Each channel is user-configurable and has two multi-purpose input/output lines, 3 external and 5 internal clock inputs. Each channel, drives an internal interrupt, which can be programmed to generate processor interrupts by using the AIC (Advanced Interrupt Controller). [10]

[46] Atmel AT91SAM7L datasheet, has used "Peripheral Data Controller" and "Peripheral DMA Controller", while pointing at the PDC. In this report, the name Peripheral DMA Controller is used.
[47] Low-profile Quad Flat Package
[48] Ball Grid Array
[49] Reset button input signal.
[50] Current Periodic Interval Value
[51] Periodic Interval Counter

The PWMC (Pulse Width Modulation Controller) module contains 4 channels. Each channel uses one of the clocks (provided from the internal clock generator) and can control one square output waveform. All channels are double buffered. This prevents the generation of an unexpected output waveform, while modifying either period or duty-cycle. [10]

The Two-wire Interface (TWI/I^2C) implemented in this chip, is made of one data line and one clock line, with up to 400 kbits/sec speed. This module supports both master and slave modes with sequential or single-byte access. It also supports multiple-master capability. [10]

Two USART modules have been implemented in this chip (USART0 and USART1), provide a full-duplex[52] synchronous or asynchronous serial communication interface. These modules support MSB- or LSB-first, 5- to 9-bit (character length) communication. They feature a programmable Baud Rate Generator. The clock source for the baud rate generator can either be the external system clock (SCK0, SCK1, SCK2) or the internal clock source. The maximum baud rate of 33400 baud (bit/sec), can be generated. Each USART channel (transmit and receive) is connected to one PDC channel. [10]

The SPI module implemented in this chip supports both master and slave modes. It provides communication between processors, in case an external processor is connected. It has four external slave chip-select lines, which provide communication with up to 15 devices. The data length is programmable from 8-bit to 16-bit. Data transmit and receive in this module is MSB first. Each channel (receive and transmit) is connected to a PDC channel. SPI interrupt line is connected to AIC (Advanced Interrupt Controller) and can be configured to be generated when data is ready for transmit or data has been received in the receive register. [10]

There is one 4-channel 10-bit SAR[53] ADC (Analog to Digital Converter) module implemented in this chip. It supports 8- or 10-bit resolution mode. The conversion result is stored in a common register (for all channels) and in a channel-dedicated register. It performs the conversion in a full range between zero volts and the reference voltage pin value. [10]

The LCD controller module implemented in this chip, is a Segment[54] LCD Controller. It is intended for monochrome passive LCD with maximum 10 common and 40 segment terminals. It is provided with its own memory and does not use the system memory. [10]

This chip includes JTAG (Joint Test Action Group), ICE (In-Circuit Emulation) and a dedicated UART debug channel (DBGU). The ARM7TDMI embedded ICE is supported via the JTAG/ICE port. The ICE provides a complete, easy to use debugging solution. Programming and debugging the the chip is possible via JTAG. The Debug Unit (DBGU) provides a UART communication that can be used for debugging purposes. This module can be used to upload a program into the internal SRAM [10]

[52] Full-duplex and half-duplex systems, provide communication between two nodes in both directions. Full-duplex system allows a simultaneous communication in both directions, while a half-duplex system allows only one direction at a time.

[53] Successive Approximate Register

[54] An LCD segment consists of two electrodes with liquid crystal between them. It becomes visible when the applied voltage is above the threshold voltage [10]

To summarize, the following is the characteristics of this microcontroller [10]:

- Performance up to 32.4 MIPS (0.9 MIPS/MHz)
- Maximum clock frequency: 36 MHz
- mA / MIPS in active state: 0.93
- 1.8 V - 3.6 V
- Maximum current consumption: 30 mA @ 36 MHz
- 80 programmable IO lines
- 10-bit SAR ADC
- No DAC
- SPI (Master/Slave support)
- I^2C(Master, Multi-Master and Slave mode support)
- Real-Time Clock (RTC)
- 3x 16-bit Timers/Counters
- 16-bit PWM Controller
- Watchdog Timer
- USB Device Controller
- DMA Controller
- On-Chip RC Oscillator
- JTAG/ICE/DBGU Interface
- LCD Controller
- Power-On-Reset
- Brown-out detector
- No USB support

Table 4.12 shows the characteristics of this microcontroller, with regards to the evaluation criteria.

Performance	Power Consumption per MIPS	I^2C	SPI	ADC	DAC
32.4 MIPS	54 mW / 32.4 MIPS = 1.67	MM/S	M/S	10-bit	-

Memory	WDT	BOR	Temperature	Packaging	Dev. Env.
128 kB	Yes	Yes	-40 to +85 °C	LQFP, LFBGA	3rd party programs

Table 4.12: AT91SAM7L128 Criteria Fulfillment
(M: Master, MM: Multi-Master, S: Slave, LQFP: Low-profile Quad Flat Package, LFBGA: Low-profile Fine-pitch Ball Grid Array)

4.1.3.6 AT91SAM7S512

This microcontroller, uses the 32-bit ARM7TDMI processor core. Figure 4.14 illustrates the functional block diagram of this microcontroller. The processor is connected to the Memory Controller module. This module manages the ASB (Advanced System Bus). APB (Advanced Peripheral Bus) is connected to this module via the Peripheral Data Controller and Peripheral Bridge. APB interfaces the processor with all peripherals. [11]

The Peripheral DMA Controller (PDC) module[55], transfers data between all on-chip serial peripheral interfaces with on-chip and external memories. This data transfer is done without processor intervention. Memory space of several peripherals has the user interface of PDC channel integrated in it. There are total number of 11 channels implemented. Most channels are implemented in pair (except for ADC). One channel in pair is dedicated to the receiving channel and the other is dedicated to transmitting channel of each peripheral (USART, DBGU, SSC and SPI). [11]

This microcontroller comes in 64-pin LQFP[56] and 64-pin QFN[57]. There are one Parallel Input/Output controllers implemented in this chip (PIOA). In total there are 32 programmable I/O lines multiplexed with up to two peripheral I/O lines. [11]

This chip contains three system timers: a Real-Time Timer (RTT), a Periodic Interval Timer (PIT) and a Watchdog Timer (WDT). The Real-Time Timer is built around a 32-bit counter and is used to count the elapsed seconds. It uses the slow clock (32.768 kHz) divided by a programmable 16-bit value and can count up to 2^{32} seconds (more than 136 years). Once it reaches a programmed value, it generates a periodic interrupt and/or triggers and alarm. [11]

The Periodic Interval Timer, provides periodic interrupts. It is built around a 20-bit CPIV[58] counter and a 12-bit PICNT[59] counter. Both counter work at the Master Clock (main peripheral clock) divided by 16. Once the counter reaches zero, it is automatically reloaded and restarted. [11]

The 12-bit Watchdog Timer is built around a 12-bit down counter. It runs on 32.768 kHz clock frequency (slow clock) and can count up to 16 seconds. It is able to generate a general reset or only a processor reset. Independent of the type of reset, Error and

[55] Atmel AT91SAM7S datasheet, has used "Peripheral Data Controller" and "Peripheral DMA Controller", while pointing at the PDC. In this report, the name Peripheral DMA Controller is used.
[56] Low-profile Quad Flat Package
[57] Quad Flat No leads
[58] Current Periodic Interval Value
[59] Periodic Interval Counter

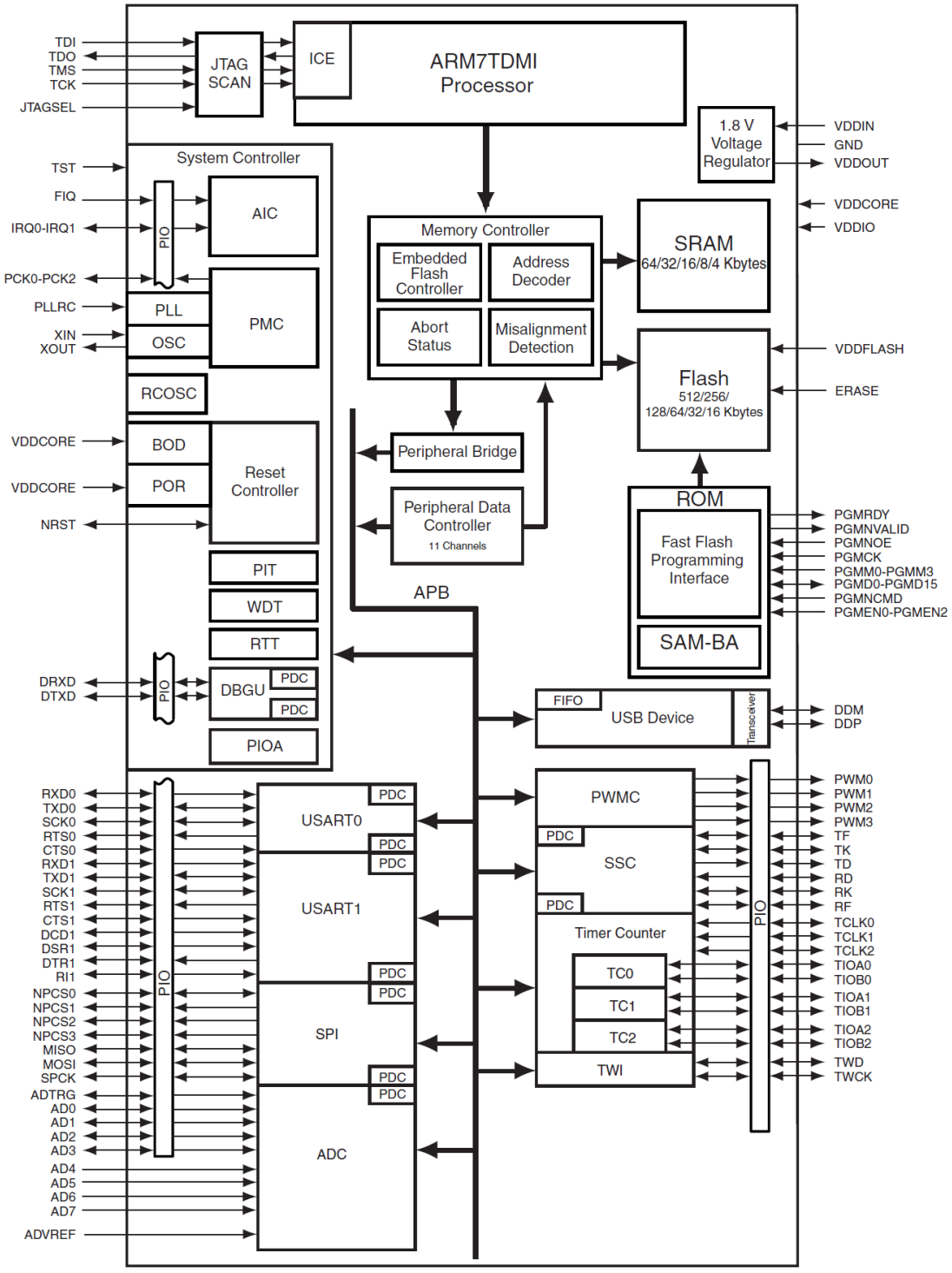


Figure 4.14: AT91SAM7S Block Diagram [11]

Underflow status bits, trigger an interrupt. The value in watchdog timer register can only be changed with a processor reset. [11]

The Advanced Interrupt Controller (AIC) module, has an 8-level priority, maskable, vectored interrupt controller. It allows the programmer to define a priority between different nIRQ (standard interrupt request) interrupt sources. This module drives the nFIQ (Fast Interrupt Request) and nIRQ inputs of the ARM processor. nFIQ line can only be asserted from the external FIQ line. The standard IRQ line delivers interrupts generated by peripherals and external interrupt requests (IRQ0 and IRQ1). [11]

The Power Management Controller (PMC) module implemented in this chip, generates all system and peripheral clocks. It also optimizes the power consumption by controlling the clocks. It is integrated with the main oscillator, slow clock (32.768 kHz) oscillator and the PLL. It provides clock for the processor (PCK), peripherals (USART, SSC, SPI, etc.) and programmable clock outputs (PCK0 to PCK2 signals). [11]

A Reset Controller module is implemented in this chip. It is based on power-on reset (POR) cells and handles all the system resets. This includes reporting which reset last occurred. This module handles the following resets (descending order): Power-up reset, Brown-out reset, Watchdog reset, Software reset and User reset. A brown-out detection circuit also implemented in it, to prevent processor from staying getting into an unpredictable state. [11]

There is one Timer/Counter block implemented in this chip. It contains three identical 16-bit timer/counter channels, which can be used for frequency measurement, interval measurement, event counting, pulse width modultaion (PWM mode) and delay timing. Each channel is user-configurable and has two multi-purpose input/output lines, 3 external and 5 internal clock inputs. Each channel, drives an internal interrupt, which can be programmed to generate processor interrupts by using the AIC (Advanced Interrupt Controller). [11]

Synchronous Serial Controller (SSC), is used to provide a synchronous communication with external devices. There are three SSC modules implemented in this chip. Each module has independent transmitter and receiver with a common clock divider. It supports many serial synchronous communication protocols (such as Long Frame Sync, Short Frame Sync, I^2S). Each module is connected to two PDC channels, to reduce processor intervention. Digital to analog conversion (DAC) is possible through these modules. [11]

The PWMC (Pulse Width Modulation Controller) module contains 4 channels. Each channel uses one of the clocks (provided from the internal clock generator) and can control one square output waveform. All channels are double buffered. This prevents the generation of an unexpected output waveform, while modifying either period or duty-cycle. [11]

The Two-wire Interface (TWI) implemented in this chip, is made of one data line and one clock line, with up to 400 kbits/sec speed. This module supports only master mode with sequential or single-byte access. It is compatible with I^2C standard. It supports 7 or 10 bit slave addressing, Standard mode speed (100 kHz) and Fast mode speed (400 kHz). [11]

This chip contains USB 2.0 Full Speed (12 Mbit/sec) device ports. it integrates an on-chip transceiver and 328-byte of configurable FIFO. Each endpoint is configurable in one

of several USB transfer types. One or two dual-port RAM (DPR) banks can be associated with this module. If two banks are associated, one will be handled with the USB Device module and the other will be handled by the processor. In isochronus endpoints, this is mandatory so the device can maintain the maximum bandwidth (1MB/s) while working with two DPR banks. [11]

The two USART modules have been implemented in this chip (USART0 and USART1), provide a full-duplex[60] synchronous or asynchronous serial communication interface. These modules support MSB- or LSB-first, 5- to 9-bit (character length) communication. USART1 is fully equipped with modem signals. Both modules feature a programmable Baud Rate Generator. The clock source for the baud rate generator can either be the external system clock (SCK0, SCK1, SCK2) or the internal clock source. The maximum baud rate of 115200 baud (bit/sec), can be generated. Each USART channel (transmit and receive) is connected to one PDC channel. [11]

The SPI module implemented in this chip supports both master and slave modes. It provides communication between processors, in case an external processor is connected. It has four external slave chip-select lines, which provide communication with up to 15 devices. The data length is programmable from 8-bit to 16-bit. Data transmit and receive in this module is MSB first. Each channel (receive and transmit) is connected to a PDC channel. SPI interrupt line is connected to AIC (Advanced Interrupt Controller) and can be configured to be generated when data is ready for transmit or data has been received in the receive register. [11]

There is one 8-channel 10-bit SAR[61] ADC (Analog to Digital Converter) module implemented in this chip. It supports 8- or 10-bit resolution mode. The conversion result is stored in a common register (for all channels) and in a channel-dedicated register. It performs the conversion in a full range between zero volts and the reference voltage pin value. [11]

This chip includes JTAG (Joint Test Action Group), ICE (In-Circuit Emulation) and a dedicated UART debug channel (DBGU). The ARM7TDMI embedded ICE is supported via the JTAG/ICE port. The ICE provides a complete, easy to use debugging solution. Programming and debugging the the chip is possible via JTAG. The Debug Unit (DBGU) provides a UART communication that can be used for debugging purposes. This module can be used to upload a program into the internal SRAM. [11]

To summarize, the following is the characteristics of this microcontroller [11]:

- Performance up to 49.5 MIPS (0.9 MIPS/MHz)
- Maximum clock frequency: 55 MHz
- 1 mA / MIPS in active state
- 1.8 / 3.3 V I/O supply, 3.3 V Flash supply, 1.8 V Core supply
- Maximum current consumption: 50 mA @ 55 MHz

[60] Full-duplex and half-duplex systems, provide communication between two nodes in both directions. Full-duplex system allows a simultaneous communication in both directions, while a half-duplex system allows only one direction at a time.

[61] Successive Approximate Register

- 32 programmable IO lines
- 10-bit SAR ADC
- DAC via SSC (Synchronous Serial Controller) modules
- SPI (Master/Slave support)
- I^2C(Master mode support)
- Real-Time Timer (RTT)
- 3x 16-bit Timers/Counters
- 16-bit PWM Controller
- Synchronous Serial Controller
- Watchdog Timer
- USB Device Controller
- DMA Controller
- On-Chip RC Oscillator
- JTAG/ICE/DBGU Interface
- Power-On-Reset
- Brown-out Detector
- No LCD Controller

Table 4.13 shows the characteristics of this microcontroller, with regards to the evaluation criteria.

Performance	**Power Consumption per MIPS**	**I^2C**	**SPI**	**ADC**	**DAC**
49.5 MIPS	165 mW / 49.5 MIPS = 3.33	M	M/S	10-bit	-

Memory	**WDT**	**BOR**	**Temperature**	**Packaging**	**Dev. Env.**
512 kB	Yes	Yes	-40 to +85 °C	LQFP, QFN	3rd party programs

Table 4.13: AT91SAM7S512 Criteria Fulfillment
(M: Master, S: Slave, LQFP: Low-profile Quad Flat Package, QFN: Quad Flat No-lead)

4.1.3.7 AT91SAM7SE512

This microcontroller, uses the 32-bit ARM7TDMI processor core. Figure 4.15 illustrates the functional block diagram of this microcontroller. The processor is connected to the Memory Controller module. This module manages the ASB (Advanced System Bus). APB (Advanced Peripheral Bus) is connected to this module via the Peripheral DMA Controller and Peripheral Bridge. APB interfaces the processor with all peripherals. [12]

The Peripheral DMA Controller (PDC) module, transfers data between all on-chip serial peripheral interfaces with on-chip and external memories. This data transfer is done without processor intervention. Memory space of several peripherals has the user interface of PDC channel integrated in it. There are total number of 11 channels implemented. Most channels are implemented in pair (except for ADC). One channel in pair is dedicated to the receiving channel and the other is dedicated to transmitting channel of each peripheral (USART, DBGU, SSC and SPI). [12]

The EBI (External Bus interface) implemented in this chip, enables data transfer between several external devices and the Memory Controller module. It features Static Memory Controller, SDRAM[62] Controller, CompactFlash® and ECC-enabled[63] NAND Flash Controller. These controllers support SRAM, PROM, EPROM, EEPROM, Flash and SDRAM, CompactFlash ® and the NAND Flash. EBI supports data transfer to up to 8 external devices. Data transfer is done through a 16-bit or 32-bit data bus and an address bus of up to 23-bits. [12]

This microcontroller comes in 128-pin LQFP (Low-profile Quad Flat Package) and 144-pin LFBGA (Low-profile Fine-pitch Ball-Grid Array). There are three Parallel Input/Output controllers implemented in this chip (PIOA, PIOB and PIOC). In total there are 88 programmable I/O lines multiplexed with up to two peripheral I/O lines. [12]

This chip contains three system timers: a Real-Time Timer (RTT), a Periodic Interval Timer (PIT) and a Watchdog Timer (WDT). The Real-Time Timer is built around a 32-bit counter and is used to count the elapsed seconds. It uses the slow clock (32.768 kHz) divided by a programmable 16-bit value and can count up to 2^{32} seconds (more than 136 years). Once it reaches a programmed value, it generates a periodic interrupt and/or triggers and alarm. [12]

The Periodic Interval Timer, provides periodic interrupts. It is built around a 20-bit CPIV[64] counter and a 12-bit PICNT[65] counter. Both counters work at the Master Clock (main peripheral clock) frequency divided by 16. Once the counter reaches zero, it is automatically reloaded and restarted. [12]

The 12-bit Watchdog Timer is built around a 12-bit down counter. It runs on 32.768 kHz clock frequency (slow clock) and can count up to 16 seconds. It is able to generate a general reset or only a processor reset. Independent of the type of reset, Error and Underflow status bits, trigger an interrupt. The value in watchdog timer register can only be changed with a processor reset. [12]

The Advanced Interrupt Controller (AIC) module, has an 8-level priority, maskable, vectored interrupt controller. It allows the programmer to define a priority between

[62] Synchronous Dynamic Random Access Memory
[63] Error Check Control
[64] Current Periodic Interval Value
[65] Periodic Interval Counter

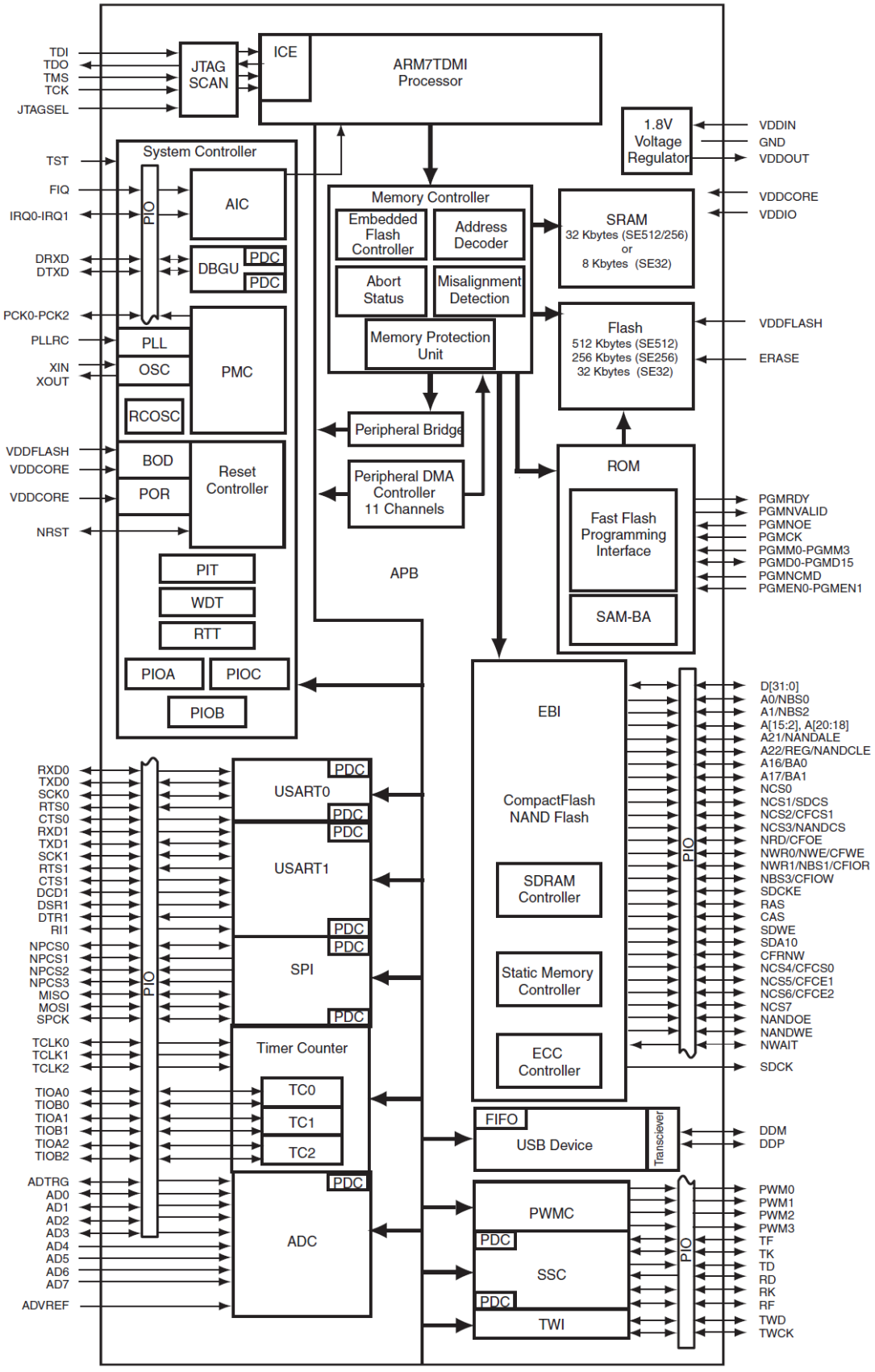


Figure 4.15: AT91SAM7SE Block Diagram [12]

different nIRQ (standard interrupt request) interrupt sources. This module drives the nFIQ (Fast Interrupt Request) and nIRQ inputs of the ARM processor. nFIQ line can only be asserted from the external FIQ line. The standard IRQ line delivers interrupts generated by peripherals and external interrupt requests (IRQ0 and IRQ1). [12]

The Power Management Controller (PMC) module implemented in this chip, generates all system and peripheral clocks. It also optimizes the power consumption by controlling the clocks. It is integrated with the main oscillator, slow clock (32.768 kHz) oscillator and the PLL. It provides clock for the processor (PCK), peripherals (USART, SSC, SPI, etc.) and programmable clock outputs (PCK0 to PCK2 signals). [12]

A Reset Controller module is implemented in this chip. It is based on power-on reset (POR) cells and handles all the system resets. This includes reporting which reset last occurred. This module handles the following resets (descending order): Power-up reset, Brown-out reset, Watchdog reset, Doftware reset and User reset. A brown-out detection circuit also implemented in it, to prevent processor from staying getting into an unpredictable state. [12]

There is one Timer/Counter block implemented in this chip. It contains three identical 16-bit timer/counter channels, which can be used for frequency measurement, interval measurement, event counting, pulse width modulation (PWM mode) and delay timing. Each channel is user-configurable and has two multi-purpose input/output lines, 3 external and 5 internal clock inputs. Each channel, drives an internal interrupt, which can be programmed to generate processor interrupts by using the AIC (Advanced Interrupt Controller). [12]

Synchronous Serial Controller (SSC), is used to provide a synchronous communication with external devices. There is one SSC module implemented in this chip. It has independent transmitter and receiver with a common clock divider. It supports many serial synchronous communication protocols (such as Long Frame Sync, Short Frame Sync, I^2S). This module is connected to two PDC channels, to reduce processor intervention. Digital to analog conversion (DAC) is possible through this modules. [12]

The PWMC (Pulse Width Modulation Controller) module contains 4 channels. Each channel uses one of the clocks (provided from the internal clock generator) and can control one square output waveform. All channels are double buffered. This prevents generation of an unexpected output waveform, while modifying either period or duty-cycle. [12]

The Two-wire Interface (TWI) implemented in this chip, is made up of one data line and one clock line, with up to 400 kbits/sec speed. This module supports both master and slave modes with sequential or single-byte access. It is compatible with I^2C standard. It supports 7- or 10-bit slave addressing, Standard mode speed (100 kHz) and Fast mode speed (400 kHz). [12]

This chip contains USB 2.0 Full Speed (12 Mbit/sec) device ports. it integrates an on-chip transceiver and 2688-byte of configurable FIFO. Each endpoint is configurable in one of several USB transfer types. One or two dual-port RAM (DPR) banks can be associated with this module. If two banks are associated, one will be handled with the USB Device module and the other will be handled by the processor. In isochronus endpoints, this is mandatory so the device can maintain the maximum bandwidth (1MB/s) while working with two DPR banks. [12]

The two USART modules have been implemented in this chip (USART0 and USART1), provide a full-duplex[66] synchronous or asynchronous serial communication interface. These modules support MSB- or LSB-first, 5- to 9-bit (character length) communication. USART1 is fully equipped with modem signals. Both modules feature a programmable Baud Rate Generator. The clock source for the baud rate generator can either be the external system clock (SCK0, SCK1) or the internal clock source. The maximum baud rate of 115200 baud (bit/sec), can be generated. Each USART channel (transmit and receive) is connected to one PDC channel. [12]

The SPI module implemented in this chip supports both master and slave modes. It provides communication between processors, in case an external processor is connected. It has four external slave chip-select lines, which provide communication with up to 15 devices. The data length is programmable from 8-bit to 16-bit. Data transmit and receive in this module is MSB first. Each channel (receive and transmit) is connected to a PDC channel. SPI interrupt line is connected to AIC (Advanced Interrupt Controller) and can be configured to be generated when data is ready for transmit or data has been received in the receive register. [12]

There is one 8-channel 10-bit SAR (Successive Approximate Register) ADC (Analog to Digital Converter) module implemented in this chip. It supports 8- or 10-bit resolution mode. The conversion result is stored in a common register (for all channels) and in a channel-dedicated register. It performs the conversion in a full range between zero volts and the reference voltage pin value. [12]

This chip includes JTAG (Joint Test Action Group), ICE (In-Circuit Emulation) and a dedicated UART debug channel (DBGU). The ARM7TDMI embedded ICE is supported via the JTAG/ICE port. The ICE provides a complete, easy to use debugging solution. Programming and debugging the the chip is possible via JTAG. The Debug Unit (DBGU) provides a UART communication that can be used for debugging purposes. This module can be used to upload a program into the internal SRAM. [12]

To summarize, the following is the characteristics of this microcontroller [12]:

- Performance up to 49.5 MIPS (0.9 MIPS/MHz)
- Maximum clock frequency: 55 MHz
- 1.6 mA / MIPS in active state
- 1.8 / 3.3 V I/O supply, 3.3 V Flash supply, 1.8 V Core supply
- Maximum current consumption: 80 mA @ 55 MHz
- 88 programmable IO lines
- 10-bit SAR ADC
- DAC via SSC (Synchronous Serial Controller) module

[66] Full-duplex and half-duplex systems, provide communication between two nodes in both directions. Full-duplex system allows a simultaneous communication in both directions, while a half-duplex system allows only one direction at a time.

- SPI (Master/Slave support)
- I^2C(Master, Multi-Master and Slave mode support)
- Real-Time Timer (RTT)
- 3x 16-bit Timers/Counters
- 16-bit PWM Controller
- Synchronous Serial Controller
- Watchdog Timer
- USB Device Controller
- DMA Controller
- On-Chip RC Oscillator
- JTAG/ICE/DBGU Interface
- Power-On-Reset
- Brown-out Detector
- No LCD Controller

Table 4.14 shows the characteristics of this microcontroller, with regards to the evaluation criteria.

Performance	**Power Consumption per MIPS**	**I^2C**	**SPI**	**ADC**	**DAC**
49.5 MIPS	144 mW / 49.5 MIPS = 2.9	MM/S	M/S	10-bit	-

Memory	**WDT**	**BOR**	**Temperature**	**Packaging**	**Dev. Env.**
512 kB	Yes	Yes	-40 to +85 °C	LQFP, LFBGA	3rd party programs

Table 4.14: AT91SAM7SE512 Criteria Fulfillment
(M: Master, MM: Multi-Master, S: Slave, LQFP: Low-profile Quad Flat Package, LFBGA: Low-profile Fine-pitch Ball Grid Array)

4.1.3.8 AT91SAM7X512

This microcontroller, uses the 32-bit ARM7TDMI processor core. Figure 4.16 illustrates the functional block diagram of this microcontroller. The processor is connected to the Memory Controller module. This module manages the ASB (Advanced System Bus). APB (Advanced Peripheral Bus) is connected to this module via the Peripheral DMA Controller and Peripheral Bridge. APB interfaces the processor with all peripherals. [14]

The Peripheral DMA Controller (PDC) module, transfers data between interfaces all on-chip serial peripherals with on-chip and external memories. This data transfer

is done without processor intervention. Memory space of several peripherals has the user interface of PDC channel integrated in it. There are total number of 13 channels implemented. Most channels are implemented in pair (except for ADC). One channel in pair is dedicated to the receiving channel and the other is dedicated to transmitting channel of each peripheral (USART, DBGU, SSC and SPI). [14]

This microcontroller comes in 64-pin LQFP (Low-profile Quad Flat Package) and 100-pin TFBGA (Thin and Fine-pitch Ball-Grid Array). There are two Parallel Input/Output controllers implemented in this chip (PIOA and PIOB). In total there are 62 programmable I/O lines multiplexed with up to two peripheral I/O lines. [14]

This chip contains three system timers: a Real-Time Timer (RTT), a Periodic Interval Timer (PIT) and a Watchdog Timer (WDT). The Real-Time Timer is built around a 32-bit counter and is used to count the elapsed seconds. It uses the slow clock (32.768 kHz) divided by a programmable 16-bit value and can count up to 2^{32} seconds (more than 136 years). Once it reaches a programmed value, it generates a periodic interrupt and/or triggers and alarm. [14]

The Periodic Interval Timer, provides periodic interrupts. It is built around a 20-bit CPIV[67] counter and a 12-bit PICNT[68] counter. Both counters work at the Master Clock (main peripheral clock) frequency divided by 16. Once the counter reaches zero, it is automatically reloaded and restarted. [14]

The Watchdog Timer is built around a 12-bit down counter. It runs on 32.768 kHz clock frequency (slow clock) and can count up to 16 seconds. It is able to generate a general reset or only a processor reset. Independent of the type of reset, Error and Underflow status bits, trigger an interrupt. The value in watchdog timer register can only be changed with a processor reset. [14]

The Advanced Interrupt Controller (AIC) module, has an 8-level priority, maskable, vectored interrupt controller. It allows the programmer to define a priority between different nIRQ (standard interrupt request) interrupt sources. This module drives the nFIQ (Fast Interrupt Request) and nIRQ inputs of the ARM processor. nFIQ line can only be asserted from the external FIQ line. The standard IRQ line delivers interrupts generated by peripherals and external interrupt requests (IRQ0 and IRQ1). [14]

The Power Management Controller (PMC) module implemented in this chip, generates all system and peripheral clocks. It also optimizes the power consumption by controlling the clocks. It is integrated with the main oscillator, slow clock (32.768 kHz) oscillator and the PLL. It provides clock for the processor (PCK), peripherals (USART, SSC, SPI, etc.) and programmable clock outputs (PCK0 to PCK3 signals). [14]

A Reset Controller module is implemented in this chip. It is based on power-on reset (POR) cells and handles all the system resets. This includes reporting which reset last occurred. This module handles the following resets (descending order): Power-up reset, Brown-out reset, Watchdog reset, Doftware reset and User reset. A brown-out detection is circuit also implemented in it, to prevent processor from getting into an unpredictable state. [14]

There is one Timer/Counter block implemented in this chip. It contains three identical 16-bit timer/counter channels, which can be used for frequency measurement, interval

[67] Current Periodic Interval Value
[68] Periodic Interval Counter

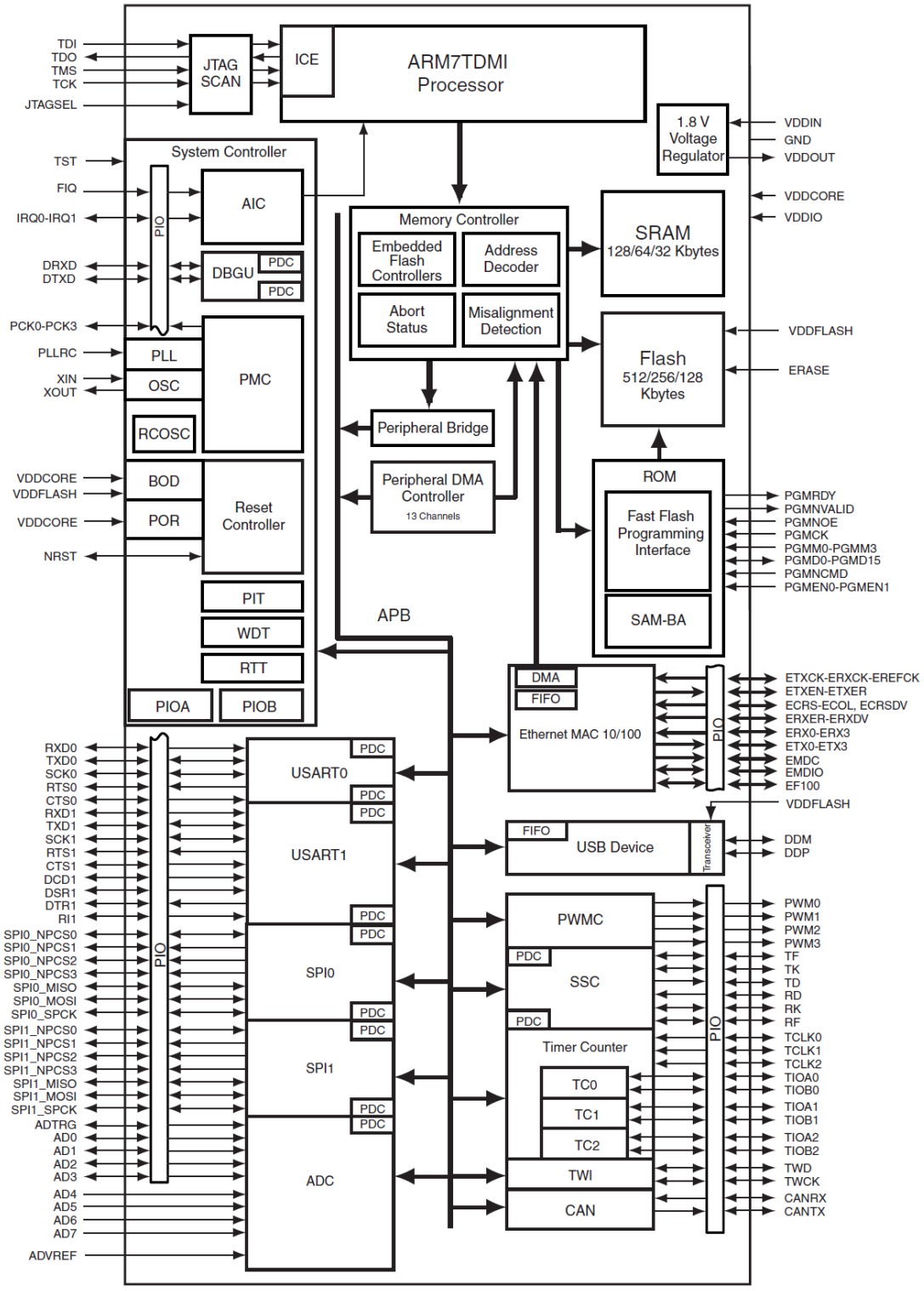


Figure 4.16: AT91SAM7X Block Diagram [14]

measurement, event counting, pulse width modulation (PWM mode) and delay timing. Each channel is user-configurable and has two multi-purpose input/output lines, 3 external and 5 internal clock inputs. Each channel, drives an internal interrupt, which can be programmed to generate processor interrupts by using the AIC (Advanced Interrupt Controller). [14]

Synchronous Serial Controller (SSC), is used to provide a synchronous communication with external devices. There is one SSC module implemented in this chip. It has independent transmitter and receiver with a common clock divider. It supports many serial synchronous communication protocols (such as Long Frame Sync, Short Frame Sync, I^2S). This module is connected to two PDC channels, to reduce processor intervention. Digital to analog conversion (DAC) is possible through this modules. [14]

The PWMC (Pulse Width Modulation Controller) module contains 4 channels. Each channel uses one of the clocks (provided from the internal clock generator) and can control one square output waveform. All channels are double buffered. This prevents generation of an unexpected output waveform, while modifying either period or duty-cycle. [14]

There is a CAN (Controller Area Network) module implemented in this chip. It provides all features necessary for implementation of CAN serial communication protocol. It has the ability to handle all frame types (Data, Remote, Error and Overload) with a bitrate of up to 1 Mbit/sec. [14]

The Two-wire Interface (TWI) implemented in this chip, is made up of one data line and one clock line, with up to 400 kbits/sec speed. This module supports both master and slave modes with sequential or single-byte access. It is compatible with I^2C standard. It supports 7- or 10-bit slave addressing, Standard mode speed (100 kHz) and Fast mode speed (400 kHz). [14]

This chip contains USB 2.0 Full Speed (12 Mbit/sec) device ports. It integrates an on-chip transceiver and 1352-byte of configurable FIFO. Each endpoint is configurable in one of several USB transfer types. One or two dual-port RAM (DPR) banks can be associated with this module. If two banks are associated, one will be handled with the USB Device module and the other will be handled by the processor. In isochronus endpoints, this is mandatory so the device can maintain the maximum bandwidth (1MB/s) while working with two DPR banks. [14]

The Ethernet MAC (EMAC) module implemented in this chip, features 10 and 100 Mbits/sec data throughput. It has a 28-byte transmit and a 28-byte receive FIFOs. This module can transfer data in two modes: Media-independent Interface (MII) and Reduced Media-independent Interface (RMII). RMII aims at reducing the pin count. EMAC features full- and half-duplex[69] operation and a DMA interface. [14]

The two USART modules have been implemented in this chip (USART0 and USART1), provide a full-duplex synchronous and asynchronous serial communication interface. These modules support MSB- or LSB-first, 5- to 9-bit (character length) communication. USART1 is fully equipped with all modem signals. Both modules feature a programmable Baud Rate Generator. The clock source for the baud rate generator

[69] Full-duplex and half-duplex systems, provide communication between two nodes in both directions. Full-duplex system allows a simultaneous communication in both directions, while a half-duplex system allows only one direction at a time.

can either be the external system clock (SCK0, SCK1) or the internal clock source. The maximum baud rate of 115200 baud (bit/sec), can be generated. Each USART channel (transmit and receive) is connected to one PDC channel. [14]

The two SPI modules implemented in this chip (SPI0 and SPI1) support both master and slave modes. They provides communication between processors, in case an external processor is connected. They have four external slave chip-select lines, which provide communication with up to 15 devices. The data length is programmable from 8-bit to 16-bit. Data transmit and receive in these module is MSB first. Each channel (receive and transmit) is connected to a PDC channel. SPI interrupt line is connected to AIC (Advanced Interrupt Controller) and can be configured to be generated when data is ready for transmit or data has been received in the receive register. [14]

There is one 8-channel 10-bit SAR (Successive Approximate Register) ADC (Analog to Digital Converter) module implemented in this chip. It supports 8- or 10-bit resolution mode. The conversion result is stored in a common register (for all channels) and in a channel-dedicated register. It performs the conversion in a full range between zero volts and the reference voltage pin value. [14]

This chip includes JTAG (Joint Test Action Group), ICE (In-Circuit Emulation) and a dedicated UART debug channel (DBGU). The ARM7TDMI embedded ICE is supported via the JTAG/ICE port. The ICE provides a complete, easy to use debugging solution. Programming and debugging the the chip is possible via JTAG. The Debug Unit (DBGU) provides a UART communication that can be used for debugging purposes. This module can be used to upload a program into the internal SRAM. [14]

To summarize, the following is the characteristics of this microcontroller [14]:

- Performance up to 49.5 MIPS (0.9 MIPS/MHz)
- Maximum clock frequency: 55 MHz
- 1.8 mA / MIPS in active state
- 1.8 / 3.3 V I/O supply, 3.3 V Flash supply, 1.8 V Core supply
- Maximum current consumption: 90 mA @ 55 MHz
- 62 programmable IO lines
- 10-bit SAR ADC
- DAC via SSC (Synchronous Serial Controller) module
- 2x SPI (Master/Slave support)
- I^2C(Master, Multi-Master and Slave mode support)
- Real-Time Timer (RTT)
- 3x 16-bit Timers/Counters
- 16-bit PWM Controller

- Synchronous Serial Controller
- Watchdog Timer
- USB Device Controller
- Ethernet MAC 10/100 Controller
- DMA Controller
- On-Chip RC Oscillator
- JTAG/ICE/DBGU Interface
- Power-On-Reset
- Brown-out Detector
- No LCD Controller

Table 4.15 shows the characteristics of this microcontroller, with regards to the evaluation criteria.

Performance	**Power Consumption per MIPS**	**I²C**	**SPI**	**ADC**	**DAC**
49.5 MIPS	162 mW / 49.5 MIPS = 3.27	MM/S	M/S	10-bit	-

Memory	**WDT**	**BOR**	**Temperature**	**Packaging**	**Dev. Env.**
512 kB	Yes	Yes	-40 to +85 °C	LQFP, TFBGA	3rd party programs

Table 4.15: AT91SAM7X512 Criteria Fulfillment
(M: Master, MM: Multi-Master, S: Slave, LQFP: Low-profile Quad Flat Package, TFBGA: Thin & Fine-pitch Ball Grid Array)

4.2 Evaluation

This section covers the evaluation of the microcontrollers. The methodology provided in Chapter 3 is applied and presented here.

4.2.1 Performance and Power Consumption

Power consumption and performance of a microcontroller are directly related to each other. Inside a microcontroller, the core (CPU), the memory storage units and the peripherals, consume the power. By changing the clock frequency of a microcontroller within its supported range, the CPU performance changes. Memory storages and peripherals use the internal peripheral clock, therefore their performance and power consumption remains the same.

Setting a microcontroller to its maximum clock frequency and activating all peripherals while reading/writing instructions and data from/to memory, draws the highest

current. This current consumption is stated in the datasheet of each microcontroller. Putting the CPU and all the peripherals to sleep or idle mode (not turning the MCU off), gives us the minimum power consumption. This number is also mentioned in each microcontroller datasheet.

In order to do a trade-off between the microcontrollers, in terms of performance, these values are put next to each other in Table 4.16. This trade-off is made to find the maximum performance of each microcontroller. The values in this table are taken from the datasheet of each microcontroller and presented here. For a more clear view over the performance of these microcontrollers, Figure 4.17 is presented. It represents the values in MIPS/MHz column of Table 4.16.

Microcontroller	**Max. Performance (MIPS)**	**Max. Clock Frequency (MHz)**	**MIPS / MHz**	**Points**
MSP430F16x	8	8	1	4
MSP430F261x	16	16	1	4
MSP430C337	4	8	0.5	1
MSP430xG461x	16	16	1	4
AT91 M42800A	30	33	0.9	3
AT91 M55800A*	-	33	-	-
AT91 RM9200	200	180	1.1	5
AT91 SAM7 A3	54	60	0.9	3
AT91 SAM7 L128	32.4	36	0.9	3
AT91 SAM7 S512	49.5	55	0.9	3
AT91 SAM7 SE512	49.5	55	0.9	3
AT91 SAM7 X512	49.5	55	0.9	3
DragonBall EZ	2.7	16.58	0.16	1
DragonBall VZ	5.4	33	0.16	1
DragonBall Super VZ	10.8	66	0.16	1

Table 4.16: Performance comparison
* Microcontroller performance is not mentioned in its datasheet.

The current consumption (highest and lowest) for each microcontroller along with the CPU voltage, are presented in Table 4.17. This trade-off is made to find the maximum power consumption of each microcontroller. The values in this table are mentioned in the datasheet of each microcontroller. Figure 4.18 illustrates the values in mW/MIPS column of Table 4.17.

Each microcontroller supports a certain clock frequency range. As it can be seen in Table 4.16, based on the maximum CPU clock frequency, the performance of microcontrollers differ from one another. For example, DragonBall Super VZ provides 10.8 MIPS[70] at 66 MHz and AT91SAM7A3 provides 54 MIPS at 60 MHz. This shows that AT91SAM7A3 has a higher performance than DragonBall Super VZ. To make the

[70]MIPS stands for Million Instructions Per Second.

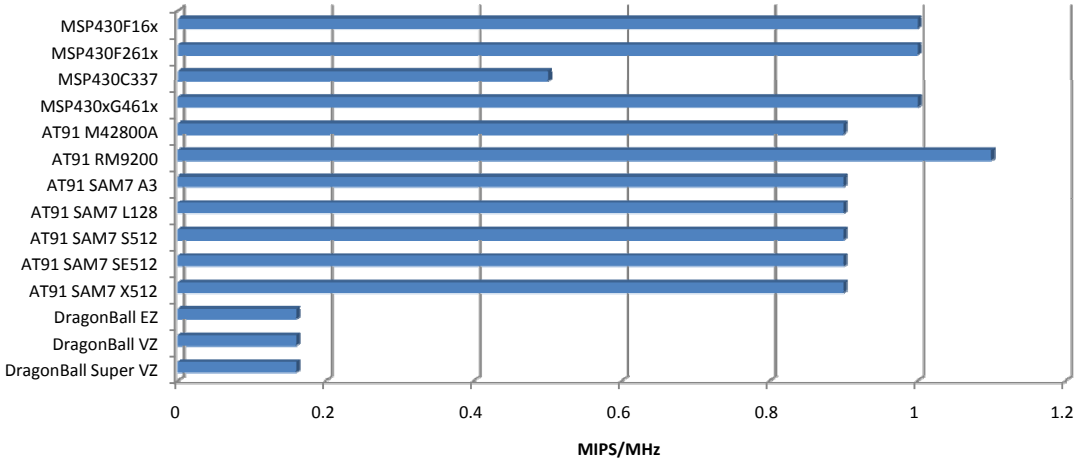


Figure 4.17: Performance comparison
(AT91M55800A is not shown in this figure.)

trade-off more clear, the performance comparison is done at the same frequency. This is acheived by calculating MIPS per MHz value (dividing the maximum performance (MIPS) by the maximum frequency (MHz) supported by each microcontroller).

As it can be seen in the MIPS/MHz column of Table 4.16, most of the microcontrollers have more or less the same performance. AT91RM9200 provides 1.1 MIPS/MHz performance, which is the highest in the table. Therefore, it receives full 5 points. MSP430F16x, MSP430F261x and MSP430xG461, each provide 1 MIPS/MHz performance. They receive 4 points. After these MCUs, come the other AT91 MCUs. These chips provide a high performance of 0.9 MIPS/MHz and each receive 3 points. MSP430C337, DragonBall EZ, DragonBall VZ and DragonBall Super VZ provide a lower performance. Therefore they receive 1 point each.

As it can be seen in Table 4.17, the core current consumption of each microcontroller (active and low power modes) is stated. In power consumption calculation, current consumption and voltage are important. Multiplying voltage and current, results in consumed power.

Maximum power consumption is achieved at the highest clock frequency supported by each microcontroller. To calculate the correct power consumption, all microcontrollers have to operate on the same task in the same amount of time. To do this, the consumed energy at 1 MIPS is calculated. This value is presented in Table 4.17 in the mW/MIPS column.

As it can be seen, AT91RM9200 has a very low power consumption of 0.2 mW/MIPS. Therefore it receives 5 points. MSP430f16x, MSP430F261x and MSP430xG461x microcontrollers are designed to be 'ultra low power' and as it can be seen, they consume around 0.7 and 0,8 mW/MIPS. In compare to AT91RM9200 power consumption, they consume a bit more and therefore they receive 4 points. AT91SAM7L128 microcontroller is also designed as a low power microcontroller. With 1.67 mW/MIPS, it does consume lower than other AT91SAM7 MCUs, but its power consumption is almost 1

Microcontroller	Active mode (mA)	Low power mode (μA)	VDD CORE	Max. Power (mW)	mW / MIPS	Points
MSP430F16x	2.640	1.1	2.2	5.808	0.726	4
MSP430F261x	5.840	0.5	2.2	12.848	0.803	4
MSP430C337	4.0	0.8	3.0	12.0	3	1
MSP430xG461x	5.920	1.0	2.2	13.024	0.814	4
AT91 M42800A*	-	-	3.0	-	-	-
AT91 M55800A*	-	-	3.0	-	-	-
AT91 RM9200	24.4	520	1.8	43.92	0.2196	5
AT91 SAM7 A3	70	175	3.3	231	4.28	0
AT91 SAM7 L128	30	3.5	1.8	54	1.67	3
AT91 SAM7 S512	50	34.3	1.8	165	3.33	1
AT91 SAM7 SE512	80	26	1.8	144	2.9	2
AT91 SAM7 X512	90	26	1.8	162	3.27	1
DragonBall EZ	20	20	3.3	66	24.4	0
DragonBall VZ	40	60	3.3	132	24.4	0
DragonBall Super VZ	80	80	3.3	264	24.4	0

Table 4.17: Typical Power Consumption Comparison
(Note: T_{Active} = 25°C, Clock Frequency = Highest supported for each MCU)
* Microcontrollers power consumption is not mentioned in its datasheet.

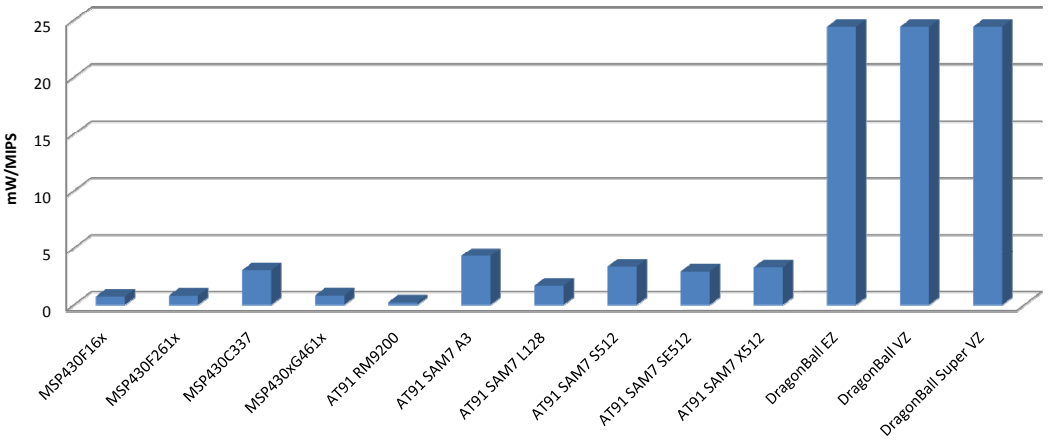


Figure 4.18: Power Consumption Comparison
(AT91M42800A and AT91M55800A are not shown in this figure.)

mW/MIPS more than the 'ultra low power' MSP430 MCUs. Therefore it receives 3 points. AT91SAM7SE512 consumes up to 2.9 mW/MIPS, which is one mW/MIPS more than the low power AT91SAM7L128 MCU. Therefore it receives 2 points. MSP430C337, AT91SAM7S512 and AT91SAM7X512, consume from 3 to less than 3.3 mW/MIPS.

These microcontrollers receive 1 point. The DragonBall microcontrollers have the highest power consumption of 24.4 mW/MIPS. In compare to the other MCUs in this table, they have a very high power consumption and receive zero points in this comparison.

To summarize, AT91RM9200 provides the highest performance and lowest power consumption among these microcontrollers. MSP430F16x, MSP430x261x and MSP430xG461x microcontrollers stand at the second place, in both performance and power consumption. The low power AT91SAM7L128 microcontroller, stands at the third place in both performance and power consumption.

4.2.2 Operating Temperature Range

As it has been mentioned before, while the satellite is orbiting Earth it frequently goes into eclipse. Based on the thermal analysis result of Delfi-C^3 (while in orbit) [25], the temperature inside the satellite changes between -10°C and 80°C. The microcontroller must tolerate this temperature range variations while active.

Microcontroller	**Min. Temp. (°C)**	**Max. Temp. (°C)**	**Score**
MSP430F16x	-40	+85	5
MSP430F261x	-40	+85	5
MSP430C337	-40	+85	5
MSP430xG461x	-40	+85	5
AT91 M42800A	-40	+85	5
AT91 M55800A	-40	+85	5
AT91 RM9200	-40	+85	5
AT91 SAM7 A3	-40	+85	5
AT91 SAM7 L128	-40	+85	5
AT91 SAM7 S512	-40	+85	5
AT91 SAM7 SE512	-40	+85	5
AT91 SAM7 X512	-40	+85	5
DragonBall EZ	0	+70	0
DragonBall VZ	0	+70	0
DragonBall Super VZ	0	+70	0

Table 4.18: Temperature Range Comparison

Table 4.18 shows the temperature range that each microcontroller supports. The values in this table are taken from the datasheet of each microcontroller. According to the datasheets, the microcontrollers can tolerate temperature variations withing this range, while they are active. If the temperature of a microcontroller goes beneath/above the minimum/maximum values stated here, device behavior becomes unpredictable.

As it can be seen in this table, all MSP430 and AT91 MCUs, support a temperature variation within -40°C and +80°C. Therefore they receive 5 points. On the other hand, the DragonBall microcontrollers do not support less than zero celcius temperature and they only support a temperature as high as +70°C. In order to use these microcontrollers, a thermal system has to be implemented in the satellite, to make sure the temperature

does not increase or decrease more than the supported range. This would create a complicated device and therefore these microcontrollers receive zero points.

4.2.3 ADC and DAC

As mentioned before, different subsystems in this satellite, use analog and digital devices (e.g. actuators, magneto-torquers, reaction-wheels, sensors, etc.). Table 4.19 displays ADC and DAC hardware support of these microcontrollers.

Microcontroller	**ADC**	**Points**	**DAC**	**Points**
MSP430F16x	12-bit SAR (8 ch.)	5	12-bit (2 ch.)	5
MSP430F261x	12-bit SAR (8 ch.)	5	12-bit (2 ch.)	5
MSP430C337	14-bit Slope	5	12-bit (2 ch.)	5
MSP430xG461x	12-bit SAR (12 ch.)	5	12-bit (2 ch.)	5
AT91 M42800A	None	0	None	0
AT91 M55800A	10-bit SAR (2×4 ch.)	5	10-bit (2×1 ch.)	5
AT91 RM9200	None	0	None	0
AT91 SAM7 A3	10-bit SAR (8 ch.)	5	None	0
AT91 SAM7 L128	10-bit SAR (8 ch.)	5	None	0
AT91 SAM7 S512	10-bit SAR (8 ch.)	5	None	0
AT91 SAM7 SE512	10-bit SAR (8 ch.)	5	None	0
AT91 SAM7 X512	10-bit SAR (8 ch.)	5	None	0
DragonBall EZ	None	0	None	0
DragonBall VZ	None	0	None	0
DragonBall Super VZ	16-bit	5	None	0

Table 4.19: ADC and DAC support comparison

The content of this table is explained in the previous section. An in-depth explanation is available in the datasheet of each microcontroller. As it can be seen, most of these microcontrollers have an ADC peripheral implemented in their hardware. The main difference is in the type of ADC (SAR or Slope) and the number of channels available in each of them.

Most of the microcontrollers that provide an ADC peripheral, have 8 channels available. In the case of AT91M55800A, there are 2 ADC modules available, which in total provides 8 channels. Each channel can be connected to one device, which means that up to 8 devices are supported by these modules. In case of MSP430xG461x, there are 12 ADC input channels available. Since these microcontrollers provide more or less similar ADC modules, they receive 5 points. The other MCUs that do not provide any ADC, receive zero points.

In terms of DAC, only a few of these microcontrollers have this module implemented in their hardware. The MSP430 DAC modules are the same as each other and provide 2 channels. AT91M55800A MCU provides two DAC modules. Each of them have only one channel available. Therefore, up to 2 devices can be supported by this chip. Since these microcontrollers provide almost similar modules, they receive 5 points each. Other microcontrollers that do not have DAC implemented in them, receive zero points.

4.2.4 I²C and SPI

As mentioned before, the main bus of this nanosatellite, is based on I²C protocol. SPI is required for communication with on-board devices (e.g. external memory, gyroscope, etc.). Table 4.20, shows I²C and SPI support of each microcontroller.

Microcontroller	I²C	Points	SPI	Points
MSP430F16x	MM/S	5	M/S	5
MSP430F261x	MM/S	5	M/S	5
MSP430C337	-	0	M/S	5
MSP430xG461x	MM/S	5	M/S	5
AT91 M42800A	-	0	M/S	5
AT91 M55800A	-	0	M/S	5
AT91 RM9200	M	0	M/S	5
AT91 SAM7 A3	M	0	M/S	5
AT91 SAM7 L128	MM/S	5	M/S	5
AT91 SAM7 S512	M	0	M/S	5
AT91 SAM7 SE512	MM/S	5	M/S	5
AT91 SAM7 X512	MM/S	5	M/S	5
DragonBall EZ	-	0	M	5
DragonBall VZ	-	0	M/S	5
DragonBall Super VZ	MM/S	5	M/S	5

Table 4.20: I²C and SPI comparison
(Supported Modes: M = Master, S = Slave, MM = Multi-Master)

In the main I²C communication of this nanosatellite, OBC[71] is set as the master node and all other subsystems are set as slave nodes. There are two OBCs on-board this satellite. The second OBC is inactive as long as the first OBC is operational. In case the first OBC fails, the second one will be activated as the master node and the satellite will continue with its mission. In case the PTRX[72] subsystem of this satellite receives a tele-command from the ground station and has to initiate this command by itself, it changes its mode to bacome the master node until the command is executed. Afterwards, it changes its mode back to become a slave node and the OBC continues the operation as the master node.

As it can be seen, seven microcontrollers provide both I²C master and slave mode support. Three microcontrollers only provide I²C functionality in master mode. The other five do not have the I²C implemented in their hardware.

The ones that support both master and slave modes, receive full 5 points. These microcontrollers can be used in any subsystem to act as both master and slave nodes on the main bus. The ones that do not support I²C at all, receive zero points. In their case, either extra coding has to be done for to implement I²C functionality in software, or an additional I²C bridge has to be implemented in each subsystem board.

[71] On-Board Computer
[72] Primary Transceiver

An I^2C bridge has I^2C on one side and digital I/O on the other side. This results in additional physical space consumption on each board. The three microcontrollers that only support master mode, also receive zero points, since most of the subsystems act as slave nodes. Using these microcontrollers also requires either adding an I^2C bridge on their boards, which uses additional space on their subsystem board or extra coding for I^2C software implementation.

As it is shown in the table, all the microcontrollers support SPI communication. Only one of them does not support SPI communication in slave mode. This is not an issue, since the devices controlled by a microcontroller are slave nodes. Therefore all the microcontrollers receive 5 points for their SPI support.

4.2.5 Watchdog Timer (WDT) and Brown-Out Reset (BOR)

As it has been mentioned before, a watchdog timer is one of the highly important circuits that a microcontrollers must have. In this section the brown-out reset circuit is also compared. WDT is necessary to ensure continues operation of a microcontroller. This timer restarts the microcontroller, in case it fails during its code operation. BOR is required to make sure the microcontroller does not malfunction or get damaged, in case of power interrupts. In case the power supply fails to provide enough power or for any reason there is a short interruption in the input power, the brown-out reset circuit will force the microcontroller to restart. This circuit is usually implemented in most of the microcontrollers being produced and is one of their safety features.

Table 4.21 shows the microcontrollers and their support for watchdog timer and brown-out reset.

Microcontroller	**Watchdog Timer**	**Points**	**Brown-out Reset**	**Points**
MSP430F16x	Y	1	Y	1
MSP430F261x	Y	1	Y	1
MSP430C337	Y	1	N	0
MSP430xG461x	Y	1	Y	1
AT91 M42800A	Y	1	N	0
AT91 M55800A	Y	1	N	0
AT91 RM9200	Y	1	N	0
AT91 SAM7 A3	Y	1	N	0
AT91 SAM7 L128	Y	1	Y	1
AT91 SAM7 S512	Y	1	Y	1
AT91 SAM7 SE512	Y	1	Y	1
AT91 SAM7 X512	Y	1	Y	1
DragonBall EZ	Y	1	N	0
DragonBall VZ	Y	1	N	0
DragonBall Super VZ	Y	1	N	0

Table 4.21: Watchdog Timer and Brown-Out Reset interface comparison (Y = Implemented on chip, N = Not implemented on chip)

In this comparison, implementation of the mentioned circuits is important. Therefore, either 1 or 0 is given as points to each microcontroller. In each case (watchdog timer implementation or brown-out reset implementation), 1 point is given to a microcontroller, if it has the circuit implemented in it. Otherwise, zero points is given to that microcontroller.

As it can be seen in the table, all microcontrollers have a watchdog timer implemented in them. Therefore all these microcontrollers receive 1 point for their WDT implementation. In terms of brown-out reset, only seven microcontrollers have this circuit implemented in them. Therefore only these microcontrollers receive 1 point. The rest of them receive zero points for not having BOR implemented.

For the microcontrollers that do not have the BOR circuit implemented, an external brown-out reset can be implemented to ensure the safety of the microcontroller.

4.2.6 Space Mission Past Experience

In case any of the microcontrollers have been used in a successful space application, there will be a higher chance of success for the Delfi-n3Xt nanosatellite. Table 4.22 shows the space application each of these microcontrollers were used in.

Microcontroller	**Space applications**	**Points**
MSP430F16x	Delfi-C^3	5
MSP430F261x	-	0
MSP430C337	-	0
MSP430xG461x	-	0
AT91 M42800A	-	0
AT91 M55800A	SwissCube	3
AT91 RM9200	-	0
AT91 SAM7 A3	-	0
AT91 SAM7 L128	-	0
AT91 SAM7 S512	-	0
AT91 SAM7 SE512	-	0
AT91 SAM7 X512	-	0
DragonBall EZ	-	0
DragonBall VZ	-	0
DragonBall Super VZ	-	0

Table 4.22: Space History Comparison

The Delfi-C^3 nanosatellite has used the MSP430F1612 and it has been working properly. Therefore, it receives 5 points for its success. According to the technical documents of the SwissCube mission, the designers selected AT91M55800A microcontroller. Therefore, it receives 3 points.

4.2.7 Memory and Packaging

As it has been mentioned before, the type of internal memory implemented in a microcontroller is important. The microcontroller size and packaging is also important. Table 4.23, shows type and size of internal memories implemented in each microcontroller, along with their packaging.

Microcontroller	Flash	/	ROM	Points	Package	Points
MSP430F16x	60 kB	/	-	5	LQFP, QFN	5
MSP430F261x	120 kB	/	-	5	LQFP, QFN	5
MSP430C337	-	/	32 kB	3	PQFP, GQFP	5
MSP430xG461x	120 kB	/	120 kB	5	LQFP	5
AT91 M42800A	-	/	-	0	LQFP, BGA	5
AT91 M55800A	-	/	-	0	LQFP, BGA	5
AT91 RM9200	-	/	128 kB	3	PQFP, BGA	5
AT91 SAM7 A3	256 KB	/	-	5	LQFP	5
AT91 SAM7 L128	128 KB	/	12 kB	5	LQFP, LFBGA	5
AT91 SAM7 S512	512 KB	/	included*	5	LQFP, QFN	5
AT91 SAM7 SE512	512 KB	/	included*	5	LQFP, LFBGA	5
AT91 SAM7 X512	512 KB	/	included*	5	LQFP, TFBGA	5
DragonBall EZ	60 KB	/	-	5	TQFP, PBGA	5
DragonBall VZ	60 KB	/	-	5	TQFP, MAPBGA	5
DragonBall Super VZ	60 KB	/	-	5	MAPBGA	5

Table 4.23: Memory and Packaging comparison
* Inetrnal ROM size is not specified in datasheet

The memory size values presented in this table, are all taken from the datasheet of each microcontroller. The packaging and number of pins for each microcontroller, have been mentioned in section 4.1.

As it can be seen from the table, most microcontrollers have internal Flash memory implemented in them. AT91M42800A and AT91M5800A have no internal memory available. These microcontrollers can boot from an external memory. Since there is no internal programming memory implemented in these microcontrollers, they receive zero points.

All AT91SAM7, DragonBall and three MSP430 microcontrollers, have internal Flash memory implemented in them. All these microcontrollers receive 5 points. MSP430C337 and AT91RM9200 only have an internal ROM memory implemented in them. These microcontrollers receive 3 points, since the data in a ROM memory cannot be updated.

All the microcontrollers come with either QFP[73] or BGA[74] packagings. These packagings are types of SMD[75] packaging. Therefore, they all receive 5 points for their low-height package types.

[73]Quad Flat Package
[74]Ball Grid Array
[75]Surface-Mount Device

4.2.8 Development Environment

As mentioned before, in order to program and debug a microcontroller, a development environment is required. However, lack of a fully featured environment does not mean that these processes cannot be done. Every company that produces a microcontroller, provides a compiler for their product.

Microcontroller	Environments	Points
MSP430F16x	CCE, CCS, CrossWorks	5
MSP430F261x	CCE, CCS, CrossWorks	5
MSP430C337	CCE, CCS, CrossWorks	5
MSP430xG461x	CCE, CCS, CrossWorks	5
AT91 M42800A	Various 3rd party programs	5
AT91 M55800A	Various 3rd party programs	5
AT91 RM9200	Various 3rd party programs	5
AT91 SAM7 A3	Various 3rd party programs	5
AT91 SAM7 L128	Various 3rd party programs	5
AT91 SAM7 S512	Various 3rd party programs	5
AT91 SAM7 SE512	Various 3rd party programs	5
AT91 SAM7 X512	Various 3rd party programs	5
DragonBall EZ	EASy68K	3
DragonBall VZ	EASy68K	3
DragonBall Super VZ	EASy68K	3

Table 4.24: Development Environment Comparison

Some of these compilers require a license for use and some of them are available for free[76]. One of the differences between the free versions and the licensed versions would be in the code size generated by them. A few compilers only generate a limited size of hex code if they are used without a license (free or trial use). The licensed version of these compilers, have no limitation in the code they generate. Another difference would be the time limitation of the free versions (trial). This section compares different types of available development environments for the microcontrollers.

According to Texas Instruments [27], they have provided a compiler (Code Composer Studio) for all the MSP430 microcontrollers. It provides capabilities such as programming, debugging and also direct memory (and register) data access to the microcontrollers.

According to Atmel, for the AT91SAM MCUs, various third party programmers are available [5]. Each programming environment provides features such as different methods to compile the codes and run-time debugging the microcontrollers.

For the DragonBall MCUs, there is no programmer dedicated to it. However, development environments designed for compiling codes and to program different microcontrollers which support this MCU can be used. Since the DragonBall is based on the Motorola 68000 CISC processor, a programmer that is used for 68000 programming can

[76] A free software is a software that can be used without restriction. These softwares are generally available without any charge.[48]

also program this microcontroller series. An open source project named EASy68k [29] is available which provides an editor, compiler and a simulator for 68000 based MCUs.

Rowley has created a development environment named CrossWorks, to program MSP430 and ARM microcontrollers. This tool is also used for programming other microcontrollers such as Maxim MAXQ developed by Dallas Semiconductor and Atmel AVR MCUs. It includes a user friendly project manager, debugger and is used to program Flash memories.

Since TI has provided a dedicated tool for programming and debugging all MSP430 (Flash or ROM based) microcontrollers, the MSP430 MCUs receive 5 points. Atmel has provided links to third party programmers. Each programmer comes with different features and they require their own licensing to be used. The AT91 MCUs also receive 5 points. Programming the DragonBall series is possible with the EASy68k tool. This program is not fully featured, but instead of a run-time debugger, it provides a simulator. The debugging is possible in the simulator, but it is not very convenient for testing external devices connected to the microcontroller. Therefore, this series of MCUs receive 3 points.

4.3 Final Result

To finalize the comparison, all the points that were calculated previously for each criteria, are gathered together in one table. Table 4.25 shows this comparison and the overall result.

As it can be seen, points of each microcontroller for every category, is shown in this table. The '*Weight*' row, represents the weight applied to (multiplied by) each point in a criteria. Sum of the result of this multiplication, is the final score of each microcontroller. This score is presented in the last column. This method has been fully described in Chapter 3.

According to the scores shown in Table 4.25, MSP430F16x has gained the highest score in compare to the other microcontrollers. In the MSP430F16x microcontroller series, there are several microcontrollers available. Therefore, a selection has to be made within them to find the most suitable microcontroller. Table 4.26 shows the main difference between these microcontrollers.

As it can be seen in Table 4.26, the only difference between these microcontrollers is in the amount of their internal Flash memory and RAM. If a microcontroller with the largest program memory (Flash memory) is chosen from the list, it may eliminate the need for a separate external memory to store data and flight software. A microcontroller with a larger amount of RAM may eliminate the need for an external RAM module. By looking at Table 4.26, it can be seen that MSP430F1611 provides maximum amount of RAM with sufficient amount of Flash memory. Since it is important for calculations to have sufficient amount of RAM available, this microcontroller is selected as the final choice.

Items	**MIPS**	**Power**	**I^{2}C**	**SPI**	**WDT**	**BOR**	**ADC**	**DAC**	**Temp**	**Space**	**Memory**	**Pkg**	**Dev. env.**	**Score**
Weight	5x	5x	5x	2x	5x	3x	4x	4x	3x	4x	4x	4x	4x	-
MSP430F16x	**4**	**4**	**5**	**5**	**1**	**1**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**218**
MSP430F261x	4	4	5	5	1	1	5	5	5	0	5	5	5	198
MSP430C337	1	1	0	5	1	0	5	5	5	0	3	5	5	132
MSP430xG461x	4	4	5	5	1	1	5	5	5	0	5	5	5	198
AT91 M42800A	3	-	0	5	1	0	0	0	5	0	0	5	5	85
AT91 M55800A	-	-	0	5	1	0	5	5	5	3	0	5	5	122
AT91 RM9200	5	5	0	5	1	0	0	0	5	0	3	5	5	132
AT91 SAM7 A3	3	0	0	5	1	0	5	0	5	0	5	5	5	125
AT91 SAM7 L128	3	3	5	5	1	1	5	0	5	0	5	5	5	168
AT91 SAM7 S512	3	1	0	5	1	1	5	0	5	0	5	5	5	133
AT91 SAM7 SE512	3	2	5	5	1	1	5	0	5	0	5	5	5	163
AT91 SAM7 X512	3	1	5	5	1	1	5	0	5	0	5	5	5	158
DragonBall EZ	1	0	0	5	1	0	0	0	0	0	5	5	3	72
DragonBall VZ	1	0	0	5	1	0	0	0	0	0	5	5	3	72
DragonBall SZ	1	0	5	5	1	0	5	0	0	0	5	5	3	117

Table 4.25: Overall microcontroller evaluation table

Microcontroller	**Flash**	**RAM**
MSP430F1610	32 KB	5 KB
MSP430F1611	*48 KB*	*10 KB*
MSP430F1612	55 KB	5 KB
MSP430F167	32 KB	1 KB
MSP430F168	48 KB	2 KB
MSP430F169	60 KB	2 KB

Table 4.26: MSP430F16x Memory Comparison

Hardware Design and Software Development

5

In order to program, test and develop the flight software for the Delfi-n3Xt nanosatellite, a test board is designed and created. This section covers the PCB (Printed Circuit Board) design and the tests performed on it.

5.1 PCB Design

To design the PCB, different PCB layout applications such as Altium Designer, CadSoft Eagle and Cadence Orcad have been looked into. These tools require licensing and can be used within a trial period[1], except for the Eagle. The Eagle can be used with either free or payed licensing and does not have a limited trial time. The only limitation with free version of the Eagle, is the board size. As it will be seen, in this step of PCB design the board size does not require to be large. Therefore Eagle is used as the PCB layout application.

5.2 Design schematics

Based on the datasheet of the MSP430F16x [44] and the libraries provided by Texas Instruments for PCB design, a schematic for this microcontroller is created. In this design, six I/O ports (48 pins) of the microcontroller are connected to pin headers. In this way it is possible to connect and use all the ports with any external device. The JTAG port of the microcontroller is connected to a 14 pin JTAG connector. These connections are based on the schematics and information provided in JTAG connection manual by Texas Instruments. More information on the JTAG is provided in section 5.3.3. Figure 5.1 shows the schematic of this board.

This microcontroller does not have a USB controller peripheral. Instead it has two sets of general I/O pins associated for serial data communication (USART0 and USART1). A FT232 microcontroller is used as the USB communication interface. According to FTDI's FT232 microcontroller datasheet, the schematics for this connection is drawn and connected to the MSP430 board design.

For the I^2C communication, a port has to be dedicated to make the connection between two boards easier. Since there are going to be more than two boards connected to each other in this satellite, two pin header ports are dedicated for this purpose. The pins on these two ports are direct map of each other. In this way, it is possible to connect infinite number of boards together and make a chain connection between them.

[1] Trial period is an amount of time that a product can be used for, without any payments. A typical trial period lasts 30 days (it may differ for each product). To use a product after the trial period is over, a license has to be acquired.

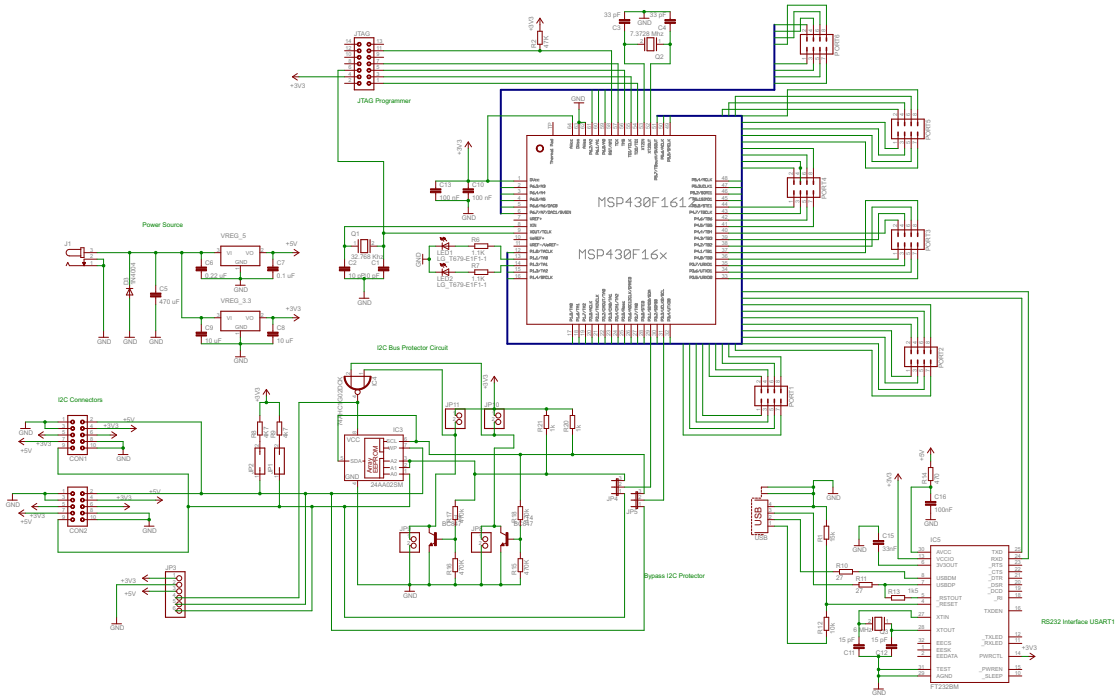


Figure 5.1: Delfi-n3Xt measurement board schematic

In order to test the I^2C communication with the I^2C protection circuit, this circuit is also implemented in this design. Between the microcontroller and the I^2C protection circuit, two jumpers are placed. These two jumpers (one for SCL and one for SDA) select the path for I^2C connection. They can either be set to connect the I^2C bus protector circuit to the I^2C line, or to skip the circuit and connect the microcontroller directly to the ports. This is useful for testing the I^2C bus protector or in case a new I^2C protector is being designed and needs to be tested.

An additional 6 pin port is dedicated to monitor the power lines and the I^2C communication. This port is placed before the two I^2C communication ports.

In addition, two LEDs are connected to pin 2 and 3 of port 1 (P1.1 and P1.2) of the microcontroller. An external 8MHz (7.328 MHz) crystal oscillator is connected to the microcontroller (XT2IN and XT2OUT). 8 MHz is the maximum clock frequency supported by this microcontroller, therefore it is chosen. The auxiliary 32.768 kHz crystal oscillator placement is also added to the design.

Two voltage regulators (3.3 volts and 5.0 volts) have also been added to the circuit to provide a steady power source to the circuit. The 3.3 volts powers up the microcontroller and the 5 volts is needed for the USB controller.

This board is designed in such a way that one external power supply connected to the power source of the boards, will provide sufficient power for all the boards. It is advised to connect one external power supply to all the boards when working with them.

This includes both times either programming the microcontroller or when running the softwares.

As it can be seen in Figure 5.1, the microcontroller I/O ports are connected via a bus to the pin headers. The I^2C connection ports are located on the left side. The I^2C bus protection circuit is located on the bottom of the schematics, between the microcontroller and the I^2C ports.

Figure 5.2 shows the PCB layout design of the Delfi-n3Xt measurement board. The microcontroller is placed in the center of the left half of the board. The six 8-pin ports are located on top and bottom of the microcontroller. Ports 1, 2 and 3 are located at the bottom and ports 4, 5 and 6 are located on the top. The JTAG connection port is located on the top left side of the board. Dual connection for the I^2C bus is located under the JTAG connector on the middle left and bottom left side of the board. Two LEDs are located right behind the lower I^2C connector. The external crystal oscillator can be placed on top of the microcontroller. The placement is very close to XTAL pins of the microcontroller for least possible noise.

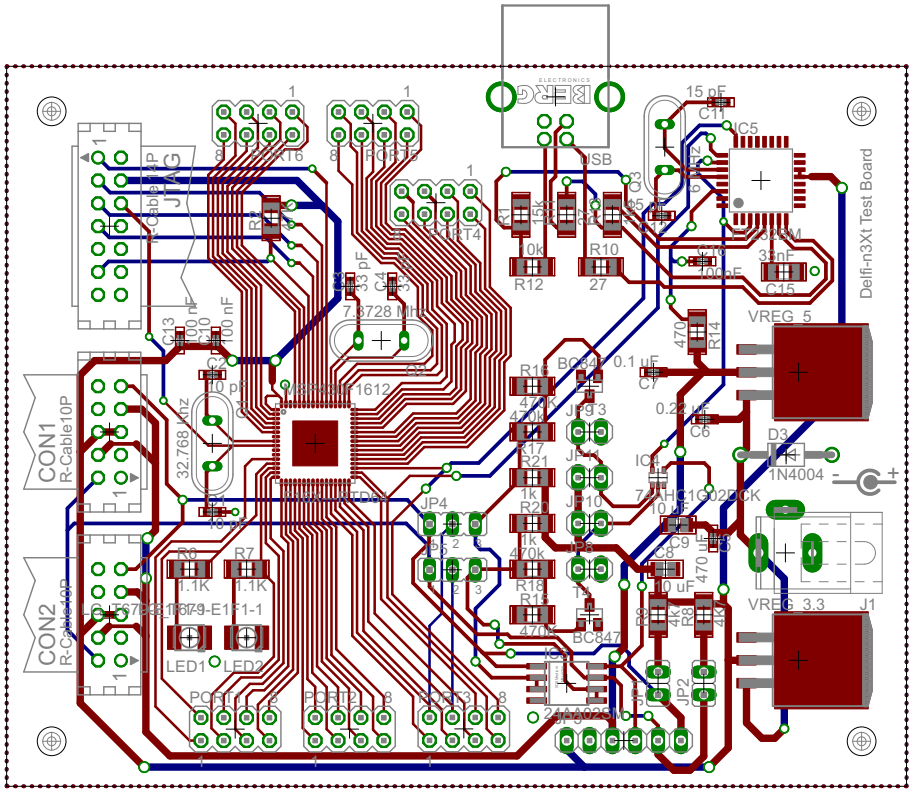


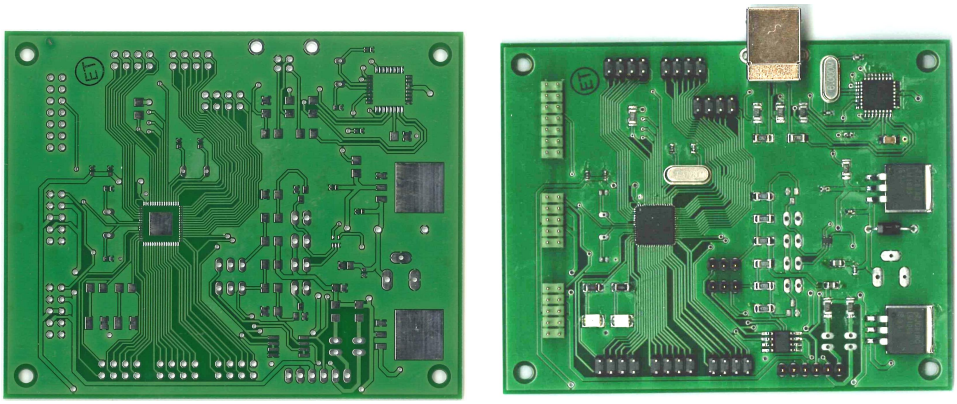
Figure 5.2: Delfi-n3Xt measurement board layout

The power source is located on the right side of the board. The voltage regulators are located on the right side to ensure that there will be no possible interference from the regulators to the chip. The external power can be connected to the input jack located between the two voltage regulators.

Between the power source and the microcontroller, the I^2C bus protection circuit is

placed. The two jumpers for enabling and disabling the bus protection circuit are located between the MCU and the bus protection circuit. The 6-pin connection dedicated for I^2C and power monitoring is located under the I^2C bus protection circuit.

The USB port and the controller (FT232) circuit is located on the top right corner of the board. The two serial lines come from the right side of the microcontroller and are connected to the USB controller. FT232 is rotated 90 degrees counter clock wise.



(a) Empty board (b) Components soldered on the board

Figure 5.3: Delfi-n3Xt measurement board

The dimensions of the board is 100mm x 80mm. Figure 5.3 shows the PCB with (5.3(b)) and without (5.3(a)) the component placed on it. All the components have been soldered on the boards by hand. In total, five boards have been created and tested.

5.3 Software Development and Tests

Information regarding the steps taken in testing the microcontroller, is presented in this section. At the beginning, basic information is provided which describe the preparation of the tests. Afterwards in section 5.3.4, the tests and their results are explained.

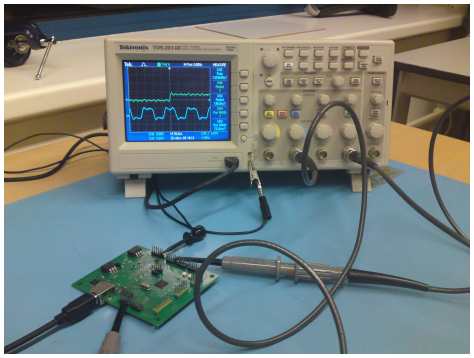
5.3.1 Development Environment

In order to begin tests and also to develop the flight software, a development environment is required. Texas Instruments provides a fully featured IDE[2], which is based on the open source Eclipse[3] IDE. The Code Composer Essentials (CCE), Code Composer Studio (CCS) and CrossWorks are three IDEs that can be used to program the MSP430 MCU. CCS and Crossworks can be used with payed license. CCE comes with free licensing.

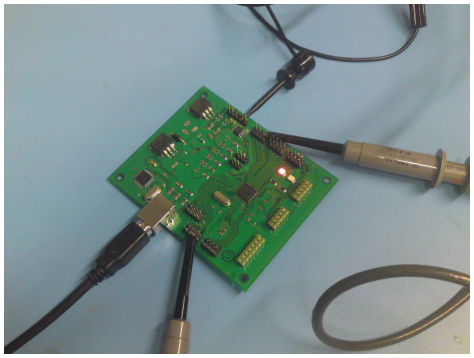
Since the CCE can be used with free licensing, it is chosen to be used in the first steps of the test. With the free licensing, there is one limitation and that is the code size.

[2]Integrated Development Environment

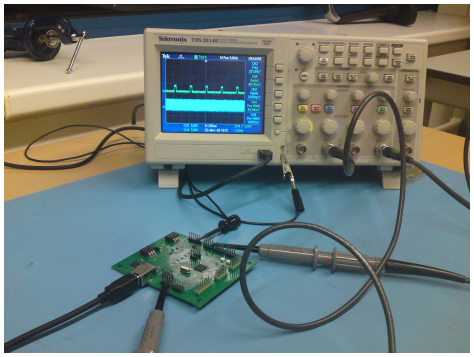
[3]Eclipse is an open source community. It is focused on building an open development platform for building, deploying and managing software. It is currently available at www.eclipse.org



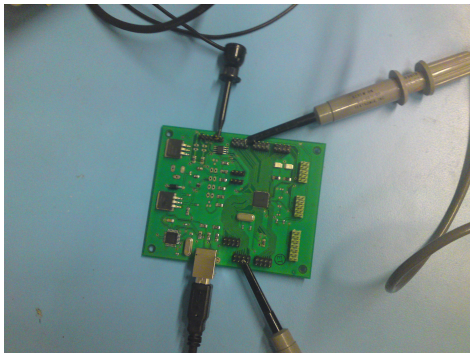
(a) Measuring the external oscillator



(b) LED blinks when a pin is set to on



(c) Testing every port while running on external clock



(d) Probes connected to the ports

Figure 5.4: Delfi-n3Xt measurement board test setup

Since the tests will not require large amount of coding, this limitation will not affect the tests.

5.3.2 Programming and Debugging

For run-time debugging the programs, JTAG interface is used. One JTAG interface can only provide communication with one microcontroller. In order to debug multiple boards altogether, multiple debuggers have to be running at the same time. In the following series of tests, only one debugger would be sufficient.

5.3.3 JTAG

JTAG is the name used for the IEEE 1149.1 standard and is an acronym for Join Test Action Group. This standard is entitled Standard Test Access Port and Boundary-Scan Architecture for test access ports used for testing PCBs using Boundary-Scan. JTAG interface consists of four/five pin interface. The following is the pins used for a JTAG connection:

- TDI (Test Data In)
- TDO (Test Data Out)

- TCK (Test Clock)
- TMS (Test Mode Select)
- TRST (Test Reset) optional

The JTAG protocol is a serial protocol (there is only one data line available). TCK is the external clock input. The TMS is used to select the operation mode. This configuration is performed by manipulating a state machine one bit at a time through the TMS signal. At every clock pulse, one bit of data is transferred in and out via TDI and TDO respectively. The TCK operating speed is usually between 10 MHz and 100 MHz. Depending on the chips, the clock speed of TCK varies. TDI data must be valid at TCK rising edge and the TDO data is valid after falling edge of the TCK.

TRST pin is an optional pin. It is an active-low reset signal to the test logic usually asynchronous (depending on the chip, it might be synchronous). In case the TRST pin is not available, the reset signal can be sent by clocking in a synchronous reset instruction.

5.3.4 Functionality Test

This section covers several tests that were run to test the board in different aspects such as reliability and communication. In some of these test, boards were used individually and in other tests they were used all together. In all these tests, one external power supply was used. In case of multiple boards connected and working together, the same power supply provides power for all the boards.

5.3.4.1 General I/O Connectivity Test

The first test would be a simple I/O test. A simple code is written which sets all the I/O pins as output and turns them on (output is set to high) and off (output is set to low) one by one. Afterwards they are turned on all together. The final step in the test turns them on in a sequence. This series of test is to check the connectivity of all the pins and also to ensure there is no cross connection between two or more pins on the board.

Figure 5.5 shows two pins of one port on the oscilloscope screen. It demonstrates the first test where the pins on one port are being turned one and off one by one. This series of tests were done on all boards and all the pins responded correctly.

5.3.4.2 Internal and External Oscillators Test

In this test a simple code is written to enable and disable usage of external crystal oscillator. The test begins with the internal RC oscillator being used to toggle the on-board LEDs. After a certain amount of time is elapsed, the external crystal oscillator is selected as the clock and the code toggles the LEDs with the external clock frequency. Both frequencies are measured with oscilloscope. This test is done to check the accuracy of the interrupts when using internal and external oscillators.

Figure 5.6 shows clock signals of one of the boards. In Figure 5.6(a) the DCO (Digitally Controlled Oscillator) is active. Figure 5.6(b) shows the MCU operating at the external 7.37280 MHz clock frequency. This test was running for one hour for every

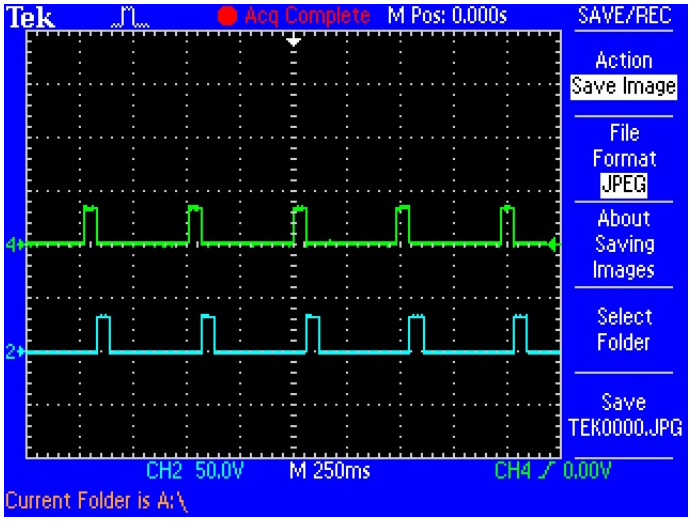
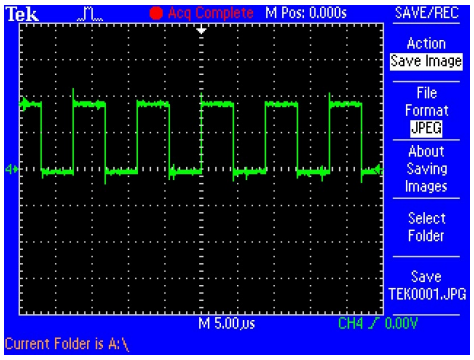
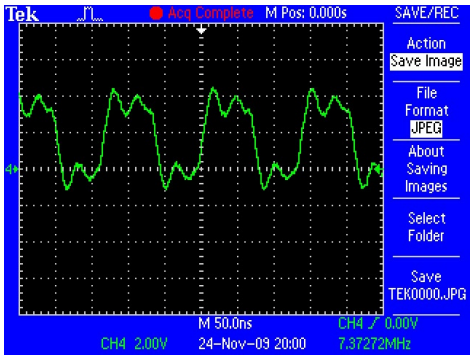


Figure 5.5: Digital I/O ports connectivity



(a) Internal DCO crystal oscillator



(b) Operating on 7.37280 MHz crystal oscillator

Figure 5.6: Internal and external oscillators test

board. In this period, the RC oscillator showed inaccurate frequencies a few times. The crystal oscillator was stable and the measured frequency was steady.

5.3.4.3 Peripheral Activity Test

To test the peripherals, several codes have been written. Each of the codes activate one peripheral and test it. The base of these codes were from code samples of MSP430 provided by Texas Instruments. Modifications were made to the code for the purpose of testing the microcontroller and the board. In these tests, all the peripherals were activated one by one and depending on the type of peripheral, different tests were made. For instance, the timers (Timer A and Timer B) were tested in all of the modes that they support (Halt, Continuous, Up, Up/Down). Watchdog timer and USARTs have also been tested. The USARTs were tested in UART, SPI and I^2C modes.

Watchdog timer was tested in both WDT and timer modes. When WDT was set as a timer, it generated interrupts at the selected time interval. With the WDT being active, two series of tests were done. One was with updating the WDT in the main code and one without updating it. In the first type of test, WDT time interval was set, the software was running, WDT was being notified that there is still no failure. To test the WDT properly, the code was modified to take longer time to execute. When the time was longer than the WDT interval, the MCU was reset and the code restarted. In the other type of test, same short code was modified and the code related to WDT was removed. This resulted in microcontroller restarting every 32 ms (the timer was set at 32 ms).

Timers were only tested with oscilloscope to see the accuracy of their interrupt. In this test a pin was being toggled based on the number given to the timer as its limit. The timers were tested in all four modes and the result was correct.

The UART, I^2C and SPI were also tested with oscilloscope for every board individually. They were also tested with two boards connected together. The data was written by one board and it was read by the other board. All these tests were successful.

6 Conclusion and Future Work

This chapter provides the summary of the work presented in this thesis and recommendation for future work.

In the design of the Delfi-n3Xt nanosatellite, power consumption and performance are the major criteria. One of the important investigations that had to be done was to find the most suitable microcontroller for this satellite in order for it to successfully accomplish its mission.

Based on the mission specifications of the Delfi-n3Xt nanosatellite and its requirements, a study has been done on the commercially available microcontrollers. A selection was made between them and based on the result, the best candidate was chosen. Therefore, an evaluation methodology was proposed. This methodology was based on the requirements of this satellite. For each requirement, a criterion was defined and a *weight* (depending on on its importance) was associated with it. Points and scores were also defined. Every microcontroller received points, based on how well it fulfilled a compared criterion. By the end, a ranking system was formed to order the microcontrollers. The goal of this system was to assign scores to each MCU based on their points, to find the most suitable microcontroller for this project.

The microcontrollers were selected and their general characteristics and specifications were discussed. Following the steps provided in the evaluation methodology, the microcontrollers were compared in each criterion, with a discussion on their differences. They got evaluated and received points. By the end, the ranking system was applied and the microcontroller with the highest score was chosen as the final microcontroller. Based on the results, the MSP430F161x microcontroller series has been chosen. In this series there are several microcontrollers available. Between these microcontrollers, an evaluation was done and MSP430F1611 was selected.

After the selection of MSP430F1611, a measurement board for this microcontroller was designed, built and tested. The test results have been briefly described. This board is widely useful for creating and testing different flight software modules for the satellite. More precisely, it includes communication within the satellite (between modules), satellite with the ground station, reading payload data within the satellite and other necessary programs.

With the rapid progress in microelectronics and computer technology, every few months several new microcontrollers become available in the market. For future development, the new microcontrollers can be evaluated, by using the methodology described in this thesis.

In addition to the work done in this thesis, the measurement board can be improved in order to make it more reliable. Codes for different modules and functionalities of this satellite can be written and tested on this board.

A modular software framework can also be developed as the base of the flight software.

Every module of the satellite can be programmed and combined to create the final codes for each satellite module. Further more, the flight software can be created based on these codes.

Bibliography

[1] Mostafa Abd-El-Barr and Hesham El-Rewini, *Fundamentals of computer organization and architecture*, John Whiley & Sons, Inc., 2005.

[2] Advanced RISC Machines Ltd (ARM), *Arm7 datasheet*, c ed., December 1994.

[3] Rakesh K. Agarwal, *80x86 architecture and programming*, Prentice-Hall, Inc., 1991.

[4] James N. Antonakos, *The 68000 microprocessor: Hardware and software principals and applications*, Pearson Prentice Hall, 2004.

[5] Atmel, *Third party flash programmers for at91sam series*, http://www.atmel.com/products/at91/thirdparty.asp#flash.

[6] Atmel, *Arm7tdmi data sheet*, b ed., January 1999.

[7] Atmel, *At91m42800a datasheet*, d ed., 2006.

[8] Atmel, *At91sam7a3 preliminary*, e ed., 2006.

[9] Atmel, *At91m55800a datasheet*, f ed., 2007.

[10] Atmel, *At91sam7l128/64 preliminary*, a ed., 2008.

[11] Atmel, *At91sam7s series preliminary*, i ed., 2008.

[12] Atmel, *At91sam7se series preliminary*, d ed., 2008.

[13] Atmel, *At91rm9200 datasheet*, i ed., 2009.

[14] Atmel, *At91sam7x series preliminary*, h ed., 2009.

[15] A. Bonnema, E.van Breukelen, and J. Rotteveel, *Delfi-C^3 requirement specification (dc3-502-010)*, Tech. report, Delft University of Technology, Faculty of Aerospace Engineering, February 2007.

[16] J. Bouwmeester, J. Go, M. de Milliano, R. Teuling, and S. de Jong, *Mission definition report of the delfi-n3xt nano-satellite mission (dnx-tud-rp-0001)*, Tech. report, Delft University of Technology, Faculty of Aerospace Engineering, April 2008.

[17] Dipto Chakravarty, *Powerpc: Concepts, architecture, and design*, McGraw-Hill, Inc., 1994.

[18] Sivarama P. Dandamudi, *Guide to risc processors - for programmers and engineers*, Springer Science+Business Media, Inc., 2005.

[19] Vibration Data, *Sputnik*, http://www.vibrationdata.com/Sputnik.htm.

[20] Sybren de Jong, *Onboard computer system design (dnx-tud-tn-0105)*, Tech. report, Delft University of Technology, Faculty of Aerospace Engineering, December 2008.

[21] Freescale Semiconductor Inc., *Mc68vz328 integrated processor user's manual*, 0 ed., 2000.

[22] Freescale Semiconductor Inc., *Mc68sz328 integrated processor reference manual*, 1.3 ed., 2004.

[23] Freescale Semiconductor Inc., *Mc68ez328 integrated processor user's manual*, 1 ed., 2005.

[24] Georgi N. Gaydadjiev and Stamatis Vassiliadis, *Introduction to computer engineering*, CE Laboratory, EWI Faculty, Delft University of Technology, September 2007.

[25] M. Graziosi, *Thermal control: Thermal analysis result*, Tech. report, Delft University of Technology, Faculty of Aerospace Engineering, Delfi-C^3, November 2007.

[26] John L. Hennessy and David A. Patterson, *Computer architecture: A quantitative approach, third edition*, Elsevier Science, June 2002.

[27] Texas Instruments, *Code composer studio integrated development environment (ide) - v4.x*, http://focus.ti.com/docs/toolsw/folders/print/ccstudio.html.

[28] Gerry Kane and Joe Heinrich, *Mips risc architecture*, Prentice-Hall, Inc., 1992.

[29] Prof. Chuck Kelly, *Easy68k home page*, http://www.easy68k.com/.

[30] F.J. Krieger, *Announcement of the first satellite, behind the sputniks*, Pravada (1957), pp. 311–12, http://history.nasa.gov/sputnik/14.html.

[31] Michael A. Miller, *The 68000 microprocessor family: Architecture, programming, and applications*, Macmillan Publishing Company, 1992.

[32] MITEL Semiconductor, *Introduction to thumb*, 3.0 ed., March 1998.

[33] Indian Space Research Organisation, *The polar satellite launch vehicle*, http://www.isro.org/Launchvehicles/PSLV/pslv.aspx.

[34] Menlo Park, *The sparc architecture manual*, Prentice-Hall, Inc., 1992.

[35] Leonid Ryzhyk, *The arm architecture*, June 2006.

[36] SSHP, *Satellite classification*, http://centaur.sstl.co.uk/SSHP/sshp_classify.html.

[37] R. Teuling, *Delfi-n3xt power budget (dnx-tud-bu-0017)*, Tech. report, Delft University of Technology, Faculty of Aerospace Engineering, September 2009.

[38] Texas Instruments, *Msp430c33x, msp430c337a datasheet*, a ed., June 2000.

[39] Texas Instruments, *Msp430x3xx family user's guide*, a ed., 2002.

[40] Texas Instruments, *Msp430x1xx family user's guide*, f ed., 2006.

[41] Texas Instruments, *Msp430x241x, msp430x261x datasheet*, a ed., June 2007.

[42] Texas Instruments, *Msp430xg461x datasheet*, g ed., October 2007.

[43] Texas Instruments, *Msp430x2xx family user's guide*, e ed., 2008.

[44] Texas Instruments, *Msp430f15x, msp430f16x, msp430f161x datasheet*, f ed., May 2009.

[45] Texas Instruments, *Msp430x4xx family user's guide*, i ed., 2009.

[46] W.J Ubbels, F.A. Mubarak, C.J.M Verhoeven, R.J. Haman, and G.L.E. Monna, *The delfi-c3 student nanosatellite - an educational test-bed for new space technology (dc3-502-010)*, Tech. report, Delft University of Technology, Faculty of Aerospace Engineering, Delfi-C^3, 2006.

[47] Wikipedia, *Arm architecture*, http://en.wikipedia.org/wiki/ARM_architecture.

[48] ———, *Free software*, http://en.wikipedia.org/wiki/Free_software.

[49] ———, *History of general purpose cpus*, http://en.wikipedia.org/wiki/History_of_general_purpose

[50] ———, *Miniaturized satellites*, http://en.wikipedia.com/wiki/Miniaturized_satellites.

[51] ———, *Modified harvard architecture*, http://en.wikipedia.org/wiki/Modified_Harvard_architectu

[52] ———, *Orthogonal instruction set*, http://en.wikipedia.org/wiki/Orthogonal_instruction_set.

[53] ———, *Satellite*, http://en.wikipedia.org/wiki/Ssatellite.

[54] ———, *Sputnik 1*, http://en.wikipedia.org/wiki/Sputnik_1.

[55] ———, *Universal serial bus*, http://en.wikipedia.org/wiki/Universal_Serial_Bus.

[56] ———, *V-2*, http://en.wikipedia.org/wiki/V-2.

MSP430 microcontrollers

A

This appendix covers the list of MSP430x1xx microcontroller series.

A.1 MSP430x1xx Summarized Characteristics

Table A.1, table A.2, table A.3 and table A.4 shows a list of all MSP430x1xx series with their specifications.

Part Number	Flash	ROM	RAM	GPIO	ADC
PMS430E112			256 B	14	Slope
MSP430P112			256 B	14	Slope
MSP430F169	60 KB		2 KB	48	12-bit SAR
MSP430F168	48 KB		2 KB	48	12-bit SAR
MSP430F167	32 KB		1 KB	48	12-bit SAR
MSP430F1612	55 KB		5 KB	48	12-bit SAR
MSP430F1611	48 KB		10 KB	48	12-bit SAR
MSP430F1610	32 KB		5 KB	48	12-bit SAR
MSP430F157	32 KB		1 KB	48	12-bit SAR
MSP430F156	24 KB		1 KB	48	12-bit SAR
MSP430F155	16 KB		512 B	48	12-bit SAR
MSP430F1491	60 KB		2 KB	48	Slope
MSP430F149	60 KB		2 KB	48	12-bit SAR
MSP430F1481	48 KB		2 KB	48	Slope
MSP430F148	48 KB		2 KB	48	12-bit SAR
MSP430F1471	32 KB		1 KB	48	Slope
MSP430F147	32 KB		1 KB	48	12-bit SAR
MSP430F135	16 KB		512 B	48	12-bit SAR
MSP430F133	8 KB		256 B	48	12-bit SAR
MSP430F1232	8 KB		256 B	22	10-bit SAR
MSP430F123	8 KB		256 B	22	Slope
MSP430F1222	4 KB		256 B	22	10-bit SAR
MSP430F122	4 KB		256 B	22	Slope
MSP430F1132	8 KB		256 B	14	10-bit SAR
MSP430F1122	4 KB		256 B	14	10-bit SAR
MSP430F1121A	4 KB		256 B	14	Slope
MSP430F1111A	2 KB		128 B	14	Slope
MSP430F1101A	1 KB		128 B	14	Slope
MSP430C1351		16 KB	512 B	48	Slope
MSP430C1331		8 KB	256 B	48	Slope
MSP430C1121		4 KB	256 B	14	Slope
MSP430C1111		2 KB	128 B	14	Slope
MSP430C1101		1 KB	128 B	14	Slope

Table A.1: MSP430x1xx series specifications

Part Number	Other Integrated Peripherals
PMS430E112	
MSP430P112	
MSP430F169	2 DAC 12 , Analog Comparator , DMA , Hardware Multiplier , SVS
MSP430F168	2 DAC 12 , Analog Comparator , DMA , Hardware Multiplier , SVS
MSP430F167	2 DAC 12 , Analog Comparator , DMA , Hardware Multiplier , SVS
MSP430F1612	2 DAC 12 , Analog Comparator , DMA , Hardware Multiplier , SVS
MSP430F1611	2 DAC 12 , Analog Comparator , DMA , Hardware Multiplier , SVS
MSP430F1610	2 DAC 12 , Analog Comparator , DMA , Hardware Multiplier , SVS
MSP430F157	2 DAC 12 , Analog Comparator , DMA , SVS
MSP430F156	2 DAC 12 , Analog Comparator , DMA , SVS
MSP430F155	2 DAC 12 , Analog Comparator , DMA , SVS
MSP430F1491	Analog Comparator , Hardware Multiplier
MSP430F149	Analog Comparator , Hardware Multiplier
MSP430F1481	Analog Comparator , Hardware Multiplier
MSP430F148	Analog Comparator , Hardware Multiplier
MSP430F1471	Analog Comparator , Hardware Multiplier
MSP430F147	Analog Comparator , Hardware Multiplier
MSP430F135	Analog Comparator
MSP430F133	Analog Comparator
MSP430F1232	
MSP430F123	Analog Comparator
MSP430F1222	
MSP430F122	Analog Comparator
MSP430F1132	
MSP430F1122	
MSP430F1121A	Analog Comparator
MSP430F1111A	Analog Comparator
MSP430F1101A	Analog Comparator
MSP430C1351	Analog Comparator
MSP430C1331	Analog Comparator
MSP430C1121	Analog Comparator
MSP430C1111	Analog Comparator
MSP430C1101	Analog Comparator

Table A.2: MSP430x1xx series specifications (continued)

Part Number	**Other Integrated Peripherals**
PMS430E112	1 16-bit (3CCR) , 1 Watchdog/Interval
MSP430P112	1 16-bit (3CCR) , 1 Watchdog/Interval
MSP430F169	1 16-bit (3CCR) , 1 16-bit (7CCR) , 1 Watchdog/Interval
MSP430F168	1 16-bit (3CCR) , 1 16-bit (7CCR) , 1 Watchdog/Interval
MSP430F167	1 16-bit (3CCR) , 1 16-bit (7CCR) , 1 Watchdog/Interval
MSP430F1612	1 16-bit (3CCR) , 1 16-bit (7CCR) , 1 Watchdog/Interval
MSP430F1611	1 16-bit (3CCR) , 1 16-bit (7CCR) , 1 Watchdog/Interval
MSP430F1610	1 16-bit (3CCR) , 1 16-bit (7CCR) , 1 Watchdog/Interval
MSP430F157	1 Watchdog/Interval , 2 16-bit (3CCR)
MSP430F156	1 Watchdog/Interval , 2 16-bit (3CCR)
MSP430F155	1 Watchdog/Interval , 2 16-bit (3CCR)
MSP430F1491	1 16-bit (3CCR) , 1 16-bit (7CCR) , 1 Watchdog/Interval
MSP430F149	1 16-bit (3CCR) , 1 16-bit (7CCR) , 1 Watchdog/Interval
MSP430F1481	1 16-bit (3CCR) , 1 16-bit (7CCR) , 1 Watchdog/Interval
MSP430F148	1 16-bit (3CCR) , 1 16-bit (7CCR) , 1 Watchdog/Interval
MSP430F1471	1 16-bit (3CCR) , 1 16-bit (7CCR) , 1 Watchdog/Interval
MSP430F147	1 16-bit (3CCR) , 1 16-bit (7CCR) , 1 Watchdog/Interval
MSP430F135	1 Watchdog/Interval , 2 16-bit (3CCR)
MSP430F133	1 Watchdog/Interval , 2 16-bit (3CCR)
MSP430F1232	1 16-bit (3CCR) , 1 Watchdog/Interval
MSP430F123	1 16-bit (3CCR) , 1 Watchdog/Interval
MSP430F1222	1 16-bit (3CCR) , 1 Watchdog/Interval
MSP430F122	1 16-bit (3CCR) , 1 Watchdog/Interval
MSP430F1132	1 16-bit (3CCR) , 1 Watchdog/Interval
MSP430F1122	1 16-bit (3CCR) , 1 Watchdog/Interval
MSP430F1121A	1 16-bit (3CCR) , 1 Watchdog/Interval
MSP430F1111A	1 16-bit (3CCR) , 1 Watchdog/Interval
MSP430F1101A	1 16-bit (3CCR) , 1 Watchdog/Interval
MSP430C1351	1 Watchdog/Interval , 2 16-bit (3CCR)
MSP430C1331	1 Watchdog/Interval , 2 16-bit (3CCR)
MSP430C1121	1 16-bit (3CCR) , 1 Watchdog/Interval
MSP430C1111	1 16-bit (3CCR) , 1 Watchdog/Interval
MSP430C1101	1 16-bit (3CCR) , 1 Watchdog/Interval

Table A.3: MSP430x1xx series specifications (continued)

Part Number	Other Integrated Peripherals
PMS430E112	
MSP430P112	
MSP430F169	1 USART (SPI or UART or I2C) , 1 USART (SPI or UART)
MSP430F168	1 USART (SPI or UART or I2C) , 1 USART (SPI or UART)
MSP430F167	1 USART (SPI or UART or I2C) , 1 USART (SPI or UART)
MSP430F1612	1 USART (SPI or UART or I2C) , 1 USART (SPI or UART)
MSP430F1611	1 USART (SPI or UART or I2C) , 1 USART (SPI or UART)
MSP430F1610	1 USART (SPI or UART or I2C) , 1 USART (SPI or UART)
MSP430F157	1 USART (SPI or UART or I2C)
MSP430F156	1 USART (SPI or UART or I2C)
MSP430F155	1 USART (SPI or UART or I2C)
MSP430F1491	2 USART (SPI or UART)
MSP430F149	2 USART (SPI or UART)
MSP430F1481	2 USART (SPI or UART)
MSP430F148	2 USART (SPI or UART)
MSP430F1471	2 USART (SPI or UART)
MSP430F147	2 USART (SPI or UART)
MSP430F135	1 USART (SPI or UART)
MSP430F133	1 USART (SPI or UART)
MSP430F1232	1 USART (SPI or UART)
MSP430F123	1 USART (SPI or UART)
MSP430F1222	1 USART (SPI or UART)
MSP430F122	1 USART (SPI or UART)
MSP430F1132	Timer UART
MSP430F1122	Timer UART
MSP430F1121A	Timer UART
MSP430F1111A	Timer UART
MSP430F1101A	Timer UART
MSP430C1351	1 USART (SPI or UART)
MSP430C1331	1 USART (SPI or UART)
MSP430C1121	Timer UART
MSP430C1111	Timer UART
MSP430C1101	Timer UART

Table A.4: MSP430x1xx series specifications (continued)

Curriculum Vitae

Armin Noroozi