# Assignment Cover Sheet

*Course Code:* **COMP2121**

*Course Name:* **microprocessors and Interfacing**

*Assignment:* **Comparing ARM and AVR architectures**

ACADEMIC REQUIREMENTS

Before submitting this assignment, the student is advised to review:

• The assessment requirements contained in the briefing document for the assignment;

• The various matters related to assessment in the course outline; and

• The *Plagiarism* and *Academic Integrity* website at https://my.unsw.edu.au/student/atoz/Plagiarism.html to ensure they are all familiar with the requirements to provide appropriate acknowledgement of source materials

If after reviewing the material there is any doubt about assessment requirements then in the first instance the student should consult with the Course Convenor and then if necessary with the Director - Undergraduate Studies.

While students are generally encouraged to work with other students to enhance learning; all assignments submitted for assessment must either be their entire own work or duly acknowledge the use of other person’s work or material. The student may be required to explain any or all parts of the assignment to the Course Convenor or other authorised persons. *Plagiarism* is using the work of others in whole or part without appropriate acknowledgement within assignment in the required form. *Collusion* is where another person(s) assists in the preparation of a student’s assignments without the consent or knowledge of the Course Convenor.

*Plagiarism* and *Collusion* are considered as Academic Misconduct and will be dealt with according to University Policy.

**Students are advised to retain a copy of this assignment for their records in case it is misplaced and has to be resubmitted. Assignments must be lodged at the Assignment Box adjacent to School Reception, Rm 159, no later than 9:00am on the assignment due date.**

• STUDENT DECLARATION OR ACADEMIC INTEGRITY

I declare that:

• This assessment item is entirely my own original work, except where I have acknowledged use of source material [such as books, journal articles, other published material, the internet, and the work of other student/s or any other person/s]; and

• This assessment item has not been submitted for assessment credit in this, or any other course, at UNSW or elsewhere.

• The assessor of this assessment item may, for the purpose of assessing this item, reproduce this assessment item and provide a copy to another member of the university.

• The assessor may communicate a copy of this assessment item to a plagiarism checking service (which may then retain a copy of the assessment item on its database for the purpose of future plagiarism checking).

*Name of student:* Abanob Tawfik

*Signature of student: AT*

*Date:* 04/04/2018

*Word Count:*

*Page Count:*

Table of Contents

[Assignment Cover Sheet 1](#_Toc510586917)

[Overview 3](#_Toc510586918)

[Pipelining 3](#_Toc510586919)

[Architecture 3](#_Toc510586920)

[Conditional execution 4](#_Toc510586921)

[Hardware support for power saving 4](#_Toc510586922)

[Caching 4](#_Toc510586923)

[Hardware support for floating point operations 4](#_Toc510586924)

[Memory models 4](#_Toc510586925)

[Memory spaces in ARM 4](#_Toc510586926)

[Purpose of memory space in ARM 4](#_Toc510586927)

[Memory size for the memory space 4](#_Toc510586928)

[Registers 5](#_Toc510586929)

[Available Registers 5](#_Toc510586930)

[General purpose register comparison 5](#_Toc510586931)

[ARM equivalent of SREG and differences 5](#_Toc510586932)

[ARM interrupt system comparison 5](#_Toc510586933)

[References 5](#_Toc510586934)

# Overview

ARM microprocessors are a type of “reduced instruction set computer” structure, also known as RISC, which require less cycles in an instruction. Having fewer cycles in instructions equates to less transistors which in turn leads to having cheaper, more power efficient machines. These qualities make ARM desirable for smaller devices which are powered by batteries including smartphones, laptops, tablets etc, and even in some cases a power efficient alternative in super computers.

### Pipelining

ARM pipeline uses a three-stage pipelining system which allows operations to perform concurrently. The three-stage system is as followed

1. Fetch - the instruction from memory
2. Decode - the registers required in the instruction
3. Execute – Registers are read from register bank, bit shifting and arithmetic is performed, Registers are written back into register bank

AVR in contrast has a two-stage pipelining system which allow for operations to be performed relatively faster than ARM

1. Fetch – the instruction from memory
2. Execute – Registers are read from register bank, bit shifting and arithmetic is performed, Registers are written back into register bank

### Architecture

As mentioned above ARM is a type of RISC architecture. In contrast AVR uses a modified Harvard 8-bit RISC Architecture. ARM architectures have the following design features.

* One cycle execution time - due to the optimised instruction set the architecture has a one clock per instruction
* Large amount of registers – to prevent a large amount of interaction with memory, RISC architectures have a large amount of registers.
* 32-bit architecture, registers from r0-r15 4 bytes (32-bits, 2 words)

AVR Architecture have the following design features.

* 8-bit architecture
* Program memory – flash memory (non-volatile)
* Data memory – SRAM, register file, input output registers.
* Registers – 32 single byte registers r0-r31, (8-bit/ half word)
* Ports – General purpose input output ports
* EEPROM – more permanent data storage, similar to flash memory it can be retained without an electrical supply.

However, both AVR and ARM architects use a load store system which splits all instructions into two parts

1. Access in memory to load or store into the desired register
2. Arithmetic operations which occur between registers

### Conditional execution

ARM architecture allows instructions to be executed conditionally, and unlike AVR which allows for conditional execution with branch and rjmp etc. ARM conditional execution can be used with most prompts. This is all done using 4 status registers, N Z C V in the instruction.

N – Negative, if the result instruction was negative

Z – Zero, if the result instruction was 0

C – Carry, if the result in instruction eg. Addition of 2 integers results in integer larger than 32 bits

V – Overflow, if the result of instruction gives signed result larger than 31 bits

These flags are either 0 or 1, and are set to 1 when condition is met. This allows for optimal conditional executing in ARM. AVR also uses a similar system with status registers.

### Hardware support for power saving

ARM processors have multiple methods of power saving which are designed into the hardware. ARM processors are in general slower than other processors which require less power. Due to having a RISC instruction set as mentioned above, this requires less transistors. Larger processes are broken down into small simpler processors which allows more work to be handled by machine code. This allows parts that aren’t being used to not use power. ARM processors also save power by going to sleep until an instruction is received. All these features in the RISC architecture alongside allowing parts to sleep unless needed, leads to power saving.

### Caching

ARM utilises a cache architecture to increase overall performance. A cache consists of both a instruction cache and data cache. The cache consists of a tag, it’s index, a word and a byte. This allows quick access to an instruction and data (instruction stored in word and data stored in byte). Caches are pieces of memory which vary, that can store temporary information.

### Hardware support for floating point operations

ARM processors have no built in floating point hardware, instead it uses an external coprocessor known as vector floating point (VFP) alongside the ARM processor. VFP supports single and double precision through the implementation of IEEE floating-point system, with 1 bit for the sign, 8 bits for the exponent and 23 bits for the fractional component.

Memory models

## Memory spaces in ARM

ARM microprocessors have only one single large memory space which can vary in size depending on the microprocessor or the usage. ARM processes split the memory space into sections for code, SRAM, peripherals, external RAM, external device, private peripherals and vendor specifics. Each of these specific features have a region in the memory space accessed via bus. This is vastly different to AVR which contains three different memory spaces.

## Purpose of memory space in ARM

Since there is only a single memory space in ARM, it is responsible for containing the code, SRAM, peripherals, external RAM, external device, private peripherals and vendor specifics. The memory space in ARM is responsible for everything. In AVR however there are three memory spaces each responsible for a different section

* the data memory – holds general purpose and i/o registers, also stores SRAM
* the program memory - holds the flash memory
* EEPROM – holds EEPROM space (similar to flash memory)

## Memory size for the memory space

The maximum memory size for the memory space in ARM is 4gb (32-bit processor).

# Registers

## Available Registers

ARM architecture contains the following 32 registers, however only 15 are available for user use.

The following registers are able to be used by user

* 13 general purpose registers R0-R12
* Stack Pointer
* Link Register

These registers are used in the back end and debugging etc.

* Program Counter
* Application Program Status Registers

## General purpose register comparison

ARM contains 13 general purpose registers, whereas AVR contains 32 general purpose registers. However, ARM registers are also 32-bit registers (2 words) whereas AVR registers are 8-bit (half words). AVR registers have two categories, general registers and i/o registers. AVR has 32 general purpose registers, R0-R31 as well as 64 i/o registers.

## ARM equivalent of SREG and differences

ARM’s equivalent to SREG is known as the Current Program Status Register (CPSR). A big difference is that the CPSR does not contain the half carry flag, also that the SREG only stores information on application program statuses. THE CPSR not only stores most of these flags but it also contains information on

* The current processor mode
* Processor state
* Endianness state
* Execution state
* Interrupt/disable flags

Below in Figure 1, is a visual representation of SREG

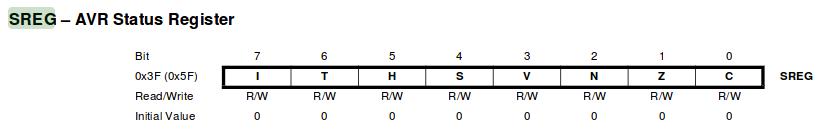


Figure 1 A visual representation of the status register

Source: http://archive.fabacademy.org/2017/fablabakgec/students/462/week8\_Embedded%20Programming/assignment8.html

Below in Figure 2, is a visual representation of CPSR

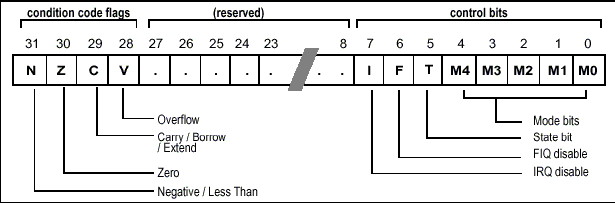


Figure 2 A visual representation of the CPSR

Source: https://stackoverflow.com/questions/21226577/arm-cpu-mode-svc-instruction

## ARM interrupt system comparison

# References

***Referenced using Harvard referencing style.***

ARM Information Center, 2018, The instruction pipeline, [ONLINE] Available from: <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0084f/ch01s01s01.html> [Accessed 03 April 2018].

Wikipedia, 2018, ARM architecture, [ONLINE] Available from: <https://en.wikipedia.org/wiki/ARM_architecture>

[Accessed 03 April 2018].

Chen C. and Novick G. and Shimano K., 2000, What is RISC?. [ONLINE] Available from: <https://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/whatis/index.html>

[Accessed 03 April 2018].

Wikipedia, 2018, ATMEL AVR, [ONLINE] Available from:

<https://en.wikipedia.org/wiki/Atmel_AVR>

[Accessed 03 April 2018].

Wikipedia,, 2018, Load/store architecture, [ONLINE] Available from:

<https://en.wikipedia.org/wiki/Load/store_architecture>

[Accessed 03 April 2018].

David Thomas, 2018, ARM: Introduction to ARM: Conditional Execution, [ONLINE] Available from: <http://www.davespace.co.uk/arm/introduction-to-arm/conditional.html>

[Accessed 03 April 2018].

ARM Information Center, 2018, Power-saving features. [ONLINE] Available from: <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dht0008a/CJAHGJEF.html>

[Accessed 03 April 2018].

What makes ARM-based chips relatively power efficient, and what is the trade-off for power consumption?, 2018, Quora, [ONLINE] Available from:

<https://www.quora.com/What-makes-ARM-based-chips-relatively-power-efficient-and-what-is-the-trade-off-for-power-consumption>

[Accessed 03 April 2018].

ARM Information Center, 2018, About cache architecture. [ONLINE] Available from: <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0201d/I21752.html>

[Accessed 03 April 2018].

ARM Information Center, 2018, Support for floating-point operations. [ONLINE] Available from:

<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0205j/CJAIDCJB.html>

[Accessed 03 April 2018].

<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dai0179b/CHDDBAEC.html>

ARM Information Center, 2018, Memory map. [ONLINE] Available from:

<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dai0179b/CHDDBAEC.html>

[Accessed 03 April 2018].

AVR tutorials, 2018, AVR Memory Map, [ONLINE] Available from:

<http://www.avr-tutorials.com/general/avr-memory-map>

[Accessed 03 April 2018].

Dynameo D., 2013, Just a few questions on ARM memory, [ONLINE] Available from:

<https://community.arm.com/tools/f/discussions/661/just-a-few-questions-on-arm-memory>

[Accessed 03 April 2018].

Arm Limited, 2005, ARM registers, [ONLINE] Available from:

<http://www.keil.com/support/man/docs/armasm/armasm_dom1359731128950.htm>

[Accessed 03 April 2018].

ARM Information Center, 2018, Current Program Status Register. [ONLINE] Available from:

<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0473c/Babdfiih.html>

[Accessed 03 April 2018].