**COMP2121**

**Assignment 1**

Lab: Monday 3-5pm

Name: Shan Wang (z5119666)

Comparing the Instruction Set Architectures of ARM and AVR

Overview

“The ARM processor is a Reduced Instruction Set Computer (RISC).” (Furber 35) The RISC principles were the main impact at the layout of the ARM processor. (Furber 20) Therefore, ARM is the development of RISC. There are an amount of salient features of ARM architecture.

Pipelining-- There exist 3-stage ARM pipeline organisation and 5-stage ARM pipeline organisation. A simple 3-stage pipeline with the following stages, fetch, decode, and execute. First of all is fetch, which is transfer the instruction from memory and then locates it in the instruction pipeline. Secondly, decode the instruction and the data path manipulates indicators organised for the following cycle. Finally, shifted the operand and generate the result and written the result into a destination register. (Furber 75) The ARM processors whose use a 5-stage pipeline which is created to satisfy a better overall performance. Stages are the following, fetch, decode, execute, buffer/data, and write-back. (Furber 79-80) Fetch and decode are similar to the 3-stage pipeline. However, in execute if the instruction is a load or save the memory address is computed inside the Arithmetic-Logic Units (ALU) instead of writing the result back. The buffer is access data memory if required, otherwise, the ALU result is virtually buffered for one clock cycle to present the equal pipeline flow for all instructions. Moreover, write-back has returned the result to the register file, containing any data loaded from memory. (Furber 79-80)

Architecture -- ARM is RISC-based architecture. (Furber 35) Furthermore, the ARM architecture incorporated plenty of features from RISC. For example, ARM is a load-store architecture, fixed-length 32-bit instructions, and 3-address instruction formats. (Furber 37) Load-store architecture means that the instruction set only handle values which are in registers and locate the outcomes into a register. (Furber 41) Fixed-length 32-bit instructions indicate that all ARM instructions are 32 bits wide. 3-address instruction formats show that the two operand registers and the register to store the result are all designated.

Conditional execution -- An unusual characteristic of the ARM instruction set is that conditional execution applies no longer only to branches however to all ARM instructions. A branch that is used to skip a small variety of following instructions can be neglected altogether by means of giving those instructions different circumstance. The advantage of conditional execution of the ARM is that it increases the speed of execution. (Furber 65)

Hardware support for power saving, caching-- “Complementary Metal Oxide Semiconductor (CMOS) is the dominant technology for current high-performance digital electronics, and has itself some good properties for low-power design.” (Furber 28) Therefore the solution of power efficiency starts from CMOS circuit. Furthermore, it by designing low-power circuit by minimising the power supply voltage, minimise the circuit activity, minimise the number of gates and minimise the clock frequency. (Furber 30) The hardware support for caching are caches, which used to faster microprocessors. It supports caching since the locality of programs which is in a certain amount of time they execute locally and repeatedly. (Furber 272)

Hardware support for floating-point operations -- The hardware support for floating-point instructions is the FPA 10 floating-point coprocessor. (Furber 162) “The ARM FPA 10 hardware floating-point accelerator supports single, double and extended double precision formats.” (Furber 165)

Memory models

In ARM system, its memory organised as linear addresses set. The use of memory is from the highest address respectively are the stack, unused area, heap, application’s image. When there is a function called, a new flame will create in the stack to store local variables and parameters and so on. And after the previous call returned it will be covered automatically by a new subroutine. Heap used to malloc a space to a new structure, and it should be freed by the user, otherwise it likely to run out of memory. The unused area depends on the stack and heap.  Therefore, the program will stop if it becomes used up. The application image is where the program memory located which a C program have access to. (Furber 180-181)

The range of application image is 1 to 4 Gbytes. To the heap and stack, the memory space depends on demand, since there is a large space between the bottom of the stack and the top of the heap. (Furber 181)

Registers

There are 15 general-purpose 32-bit registers (r0 to r14), the program counter (r15) and the current program status register (CPSR). The registers above are needed to be considered when writing a user-level program. (Furber 40) Instruction register (IR) that keep the current instruction while it is completed. (Furber 7) Program counter (PC) register which is used to preserve the address of the current instruction. The CPSR is used in user-level programs to keep the condition code bits. (Furber 40)

In ARM, general purpose registers are r0 to r14, and the instruction set provides complete access to r0 to r7 registers. Furthermore, r13 to r15 used for a special purpose, for example, r13 is used as a stack pointer, and r14 is the link register, and r15 is the program counter (PC). Moreover, the rest registers which include r8 to r12 and CPSR are used with the limit. (Furber 190) However, in AVR, general-purpose registers are 32 8-bit wide registers from r0 to r31. r26 to r31 can be used as three 16-bit indirect address register pointer for data space addressing. These registers can be describe as X(r27:r26), and Y(r29:r28), and Z(r31:r30). (*Atmega640/1280/1281/2560/2561 - Complete* 13,15-16) Furthermore, r0 to r15 have less access than r16 to r31.

The CPSR in ARM is equivalent to the SREG in AVR. In general, the width of these two registers is different and some purposes are same. And ARM has less purpose bits than AVR. In ARM, the CPSR is 32-bit register. The 4 bits at the bottom of the register hold the processor mode. The instruction set (T) at the fifth bit and interrupt enables (I and F) at the sixth and seventh bit. These bits cannot be changed in user-level. The most significant four bits of the register are N, Z, C, and V. These bits have the following meaning. N stand for Negative, the flag will be changed when producing a negative result. Z means Zero, the flag is set when the result came out is zero. C is Carry, if there is a carry-out, the flag is on. V is oVerflow, the flag will be on while generating an overflow into the sign bit. (Furber 40) The remaining bit of the register is unused. However, in AVR, the SREG is 8-bit register. From high significant bits to low significant bits respectively are I, T, H, S, V, N, Z, and C. I stand for Global Interrupt Enable when the interrupt is enabled this bit must be set. T means Bit Copy Storage, its use as a location to store the bit. H is Half Carry Flag, which indicates a half carry in some arithmetic operations. S is Sign Bit, which always an exclusive or between the N and the V. N stands for Negative Flag, which shows whether the result of the operation is positive or negative. Z is Zero Flag, it indicates if the outcome of the operation is zero. C is Carry Flag, which means there is a carry-in operation. (*Atmega640/1280/1281/2560/2561 - Complete* 14-15)

The interrupt system in ARM is called Software Interrupt (SWI). The SWI is used for ‘supervisor call’, which means the processor is placed in supervisor mode and start executing instructions from 0x08. The actions of the processor firstly store the instruction address which after the SWI in r14\_svc. Secondly, place CPSR in SPSR\_svc. After that, enter supervisor mode and disable IRQs (Interrupt ReQuests). Finally, set the PC to 08 and execute the instructions. In order to return after finishing the SWI system routine, the value in r14\_svc must be reloaded to PC as well as copy SPSR\_svc back to the CPSR. (Furber 117) In AVR, it has several interrupt sources. To enable the interrupt, individual enable bits and the Global Interrupt Enable bit in the SREG(Status Register) must be set by writing logic one. After an interrupt happens, all interrupt are disabled by clearing the I-bit of the Global Interrupt Enable. While the I-bit can be set if there are nested interrupts. When the routine is returned, the I-bit is set automatically. Generally, there are two types of the interrupt, the first type is started by set the interrupt flag on. And the second type will begin as long as the interrupt condition exists. As the ARM, in AVR, the SREG always needed to store before the routine and restore after it returns. Finally, the interrupts must be disabled. (*Atmega640/1280/1281/2560/2561 - Complete* 18-19)

Instruction Set

ARM instructions are all 32-bit words and must be word-aligned. However, there exist a subset which is encoded into 16-bit instructions which are called Thumb instructions and these instructions are half-words and must be aligned on 2-byte boundaries. (Furber 106) In AVR, most of the instructions have a single 16-bit word format. And all program memory address includes a 16-bit or 32-bit instruction.

Branch and link instructions are used for the branch to a subroutine in a way that makes it likely to resume the original code sequence when the subroutine has completed.the branch and link instruction which is as branch instructions and stores the address of the instruction following the branch in r14 which is the link register. Furthermore, since the return address is saved in r14, the subroutine cannot be called without saving of r14. And a normal way to fix it is to use a stack. (Furber 66)

The instructions used for stack in ARM is stack pointer which is r13 and it used for keep the address of the top of the stack by pointing to the last value pushed onto the stack or the next slot where the next data will be placed. Moreover, the stack pointer needs to be initialised first. In AVR, the stack pointer register is SP, which points to the top of the Stack as ARM. The stack pointer must be set above 0x200. When data pushed into the stack the stack pointer decremented and incremented when values are popped. Furthermore, in AVR, the stack pointer is as two 8-bit registers in I/O space. (*Atmega640/1280/1281/2560/2561 - Complete* 16)

In ARM, since it is difficult to read and write data correctly, therefore, I/O locations need to be accessed through supervisor calls (SWIs) or by C library functions to use those calls. (Furber 312) In AVR, I/O registers in the address between 0x00 and 0x1F can bit-accessible by using the SBI and CBI instructions and the bit value can be checked by the SBIS and SBIC instructions. LD and ST are used for addressing registers as data space. (*Atmega640/1280/1281/2560/2561 - Complete* 27)

In ARM, addressing modes which are available with ARM load/store instructions are Immediate addressing, register addressing, since, in load-store architecture, the instruction set will only process values which are stored in registers. (Furber 17 and 41) In AVR, there are five addressing modes, direct, indirect with displacement, indirect with pre-decrement, indirect with post-increment. And all these modes are not available with load-store instructions. (*Atmega640/1280/1281/2560/2561 - Complete* 22)

Data types

ARM processors support the following six data types: 8-bit signed and unsigned bytes, 16-bit signed and unsigned half-words, and 32-bit signed and unsigned words. Since the ARM operations are 32-bit instructions and the width of registers is 32 bits, therefore, the data types that can be natively represented by ARM’s registers are 32-bit signed and unsigned words. (Furber 106)

ARM instruction set can operate on 64-bit integers. A 64-bit addition can be calculated by two 32-bit additions, C flag in the Status register is used to transfer the carry bit from lower words to higher words. The instruction related is ADC which means add high with carry, and ADD stand for addition. (Furber 155-156)

Conclusion

The advantage of ARM is it allows greater complexity since it higher functionality, the drawback of ARM is that it has higher manufacturing cost. ARM is suited to large-scale embedded applications, for example, industrial automation. The positive aspect of AVR is the low power consumption and money consumption. And AVR is suited to smaller applications, for instance, home appliances and energy-efficiency lighting.

Reference

*Atmega640/1280/1281/2560/2561 - Complete*. 1st ed. Atmel Corporation, 2014. Web. 2 May 2017.

Furber, Steven B. *ARM System-On-Chip Architecture*. 1st ed. London: Addison-Wesley, 2000. Print.