**Introduction**

The ARM ISA (ARM Instruction Set Architecture) is an instruction set developed from the MIPS ISA. Both ISAs are RISC architectures but ARM ISA tends to be much more complicated than their simpler counterpart. It is also worthed noting that ARM ISA is a proprietary architecture and requires licenses unlike MIPS. It is also worth noting that ARM ISA uses little Endian and MIPS uses Big Endian. This report will mainly discuss about ARM Assembly.

Reference: https://www.slideshare.net/PriyangaKR1/arm-instruction-set-239425383

**Instructions Classes**

In general, ARM ISA contains very similar instruction classes to the MIPS assembly as expected. There are a total of 3 basic instruction classes in ARM ISA. Data Processing Instructions (Arithmetic and Logical Operators), Data Transfer Instructions (Load Word, Store Word) and Control Flow Instructions (Branch Control, Conditionals and Jump Statements).

1. Data Processing Instructions

* All operands are 32 bits and all result registers are specified in the instruction. Very similar to MIPS.
* There is one exception to the said rule, long multiply.

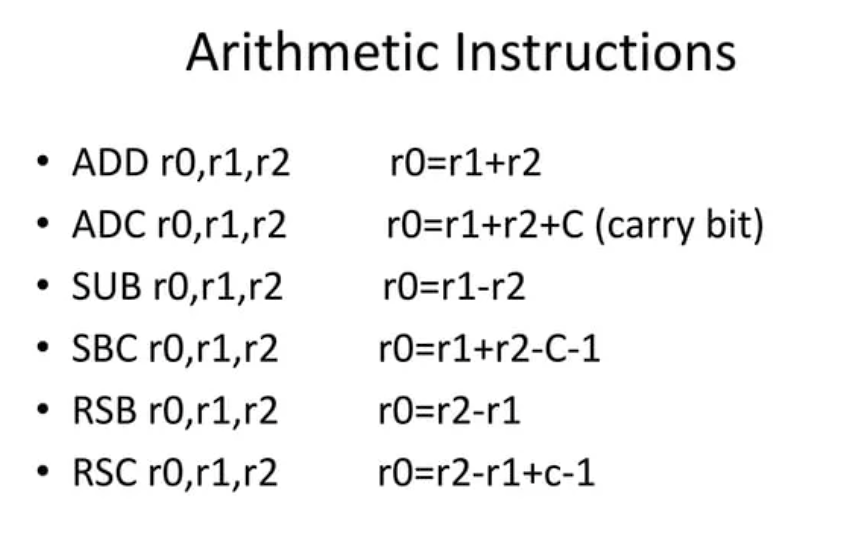


Figure 1: Some arithmetic instructions of the ARM Assembly.

* The ADD and SUB instructions works exactly like MIPS Assembly.
* The ADC and SBC instructions (Add with Carry and Sub with carry) is similar to the regular ADD and SUB instructions but with a with a “Carry Bit” added and subtracted respectively. The instruction is used and written the same as ADD and SUB.
* The RSB or (Reverse sub) and RSC (Reverse sub with carry) works similar to the regular. The only key differences are that “r1” and “r2” placements are different. The equivalent of RSB r0, r1, r2 is SUB and RSB. The similar could also be said for RSC
* It is also worthed noting that instead of a register value, a constant immediate could also be used in these operation, similar to MIPS Assembly.

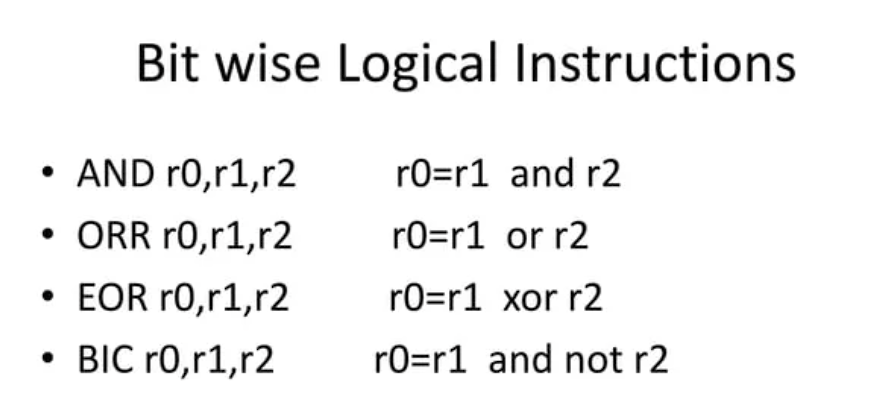


Figure 2: Some Logical Instructions in ARM Assembly.

* The AND instruction is the same as MIPS Assembly. The ORR instruction is also the same as the one in MIPS but the spelling is not the same. The formatting is also exactly the same.
* The EOR instruction is the new one, this one represents the XOR logical operand. The usage is similar to the regular ORR instruction.
* The BIC or the “Bit Clear” instruction is the new one. Sometimes this instruction is called “AND OR” instructions. It negates the second register (r2) and does an AND operation and store the result in r0.

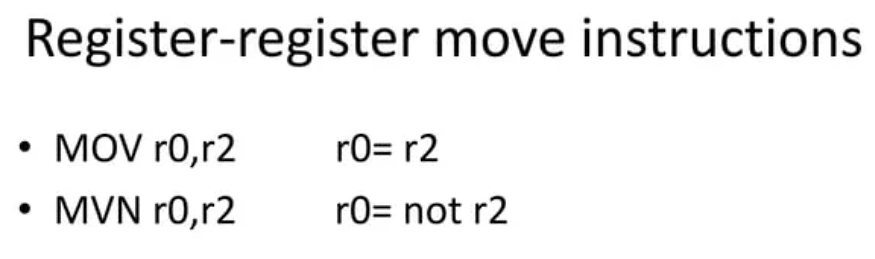


Figure 3: Register to Register instruction move instructions in ARM Assembly.