

# Computer Architecture

과제 #1: ARM Instructions 분석

2021년 1학기  
Young Geun Kim (김영근)

# 1차 과제 목적

- 하드웨어 명령어 및 Assembly 언어의 이해
- ARM Reference Manual을 통한 ARM Instruction 숙지
- 시스템 및 하드웨어 레벨에서 명령어가 어떤 식으로 수행되는지 숙지

# 1차 과제 내용

- Instruction File (inst\_data.mif) 내 000번 ~ 024번 주소의 ARM 명령어 분석
  - 16진수로 되어 있는 명령어를 2진수 (Binary)로 변환
  - 어떤 명령어인지 ARM Reference Manual을 통해 확인
  - 해당 명령어가 어떤 의미를 가지는지 서술
  - 분석한 의미에 맞게 실제 동작 순서 기록
  - 동작이 어디서 끝나는지 명시  
(끝나지 않는다면, 어디서부터 어디까지 반복되는지 명시)

# inst\_data.mif

address  
(16진수)

```
000 : EA000006;  
001 : EAffffff;  
002 : EA0000A7;  
[003..005] : EAffffff;  
006 : EA0000A4;  
007 : EAffffff;  
008 : E59F2EC8;  
009 : E3A00040;  
00A : E5820010;  
00B : E5820014;  
00C : E5820018;  
00D : E582001C;  
00E : E5820020;  
00F : E5820024;  
010 : E3A0003F;  
011 : E5820028;  
012 : E3A00008;  
013 : E582002C;  
014 : E59F3E9C;  
015 : E59F1E9C;  
016 : E5831000;  
017 : E59F9E98;  
018 : E3A08000;  
019 : E5898000;  
01A : E5898004;  
01B : E5898008;  
01C : E589800C;  
01D : E5898010;  
01E : E5898014;  
01F : E5898018;  
020 : E59FDE78;  
021 : E5931200;  
022 : E3510001;  
023 : 0A000000;  
024 : EAffffff;
```

EA000006

instruction  
(16진수)

# arm\_architecture\_reference\_manual.pdf

## A3.1 Instruction set encoding

Figure A3-1 shows the ARM instruction set encoding.

All other bit patterns are UNPREDICTABLE or UNDEFINED. See *Extending the instruction set* on page A3-32 for a description of the cases where instructions are UNDEFINED.

An entry in square brackets, for example [1], indicates that more information is given after the figure.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data processing immediate shift	cond [1]	0	0	0	opcode			S	Rn			Rd			shift amount			shift	0	Rm												
Miscellaneous instructions: See Figure A3-4	cond [1]	0	0	0	1	0	x	0	x x x x x x x x x x x x x x x x													0	x x x x x									
Data processing register shift [2]	cond [1]	0	0	0	opcode			S	Rn			Rd			Rs			0	shift	1	Rm											
Miscellaneous instructions: See Figure A3-4	cond [1]	0	0	0	1	0	x	0	x x x x x x x x x x x x x x x x													0	x	x	1	x x x x x						
Multiplies: See Figure A3-3 Extra load/stores: See Figure A3-5	cond [1]	0	0	0	x	x	x	x	x	x x x x x x x x x x x x x x x x													1	x	x	1	x x x x x					
Data processing immediate [2]	cond [1]	0	0	1	opcode			S	Rn			Rd			rotate			immediate														
Undefined instruction	cond [1]	0	0	1	1	0	x	0	0	x x x x x x x x x x x x x x x x																						
Move immediate to status register	cond [1]	0	0	1	1	0	R	1	0	Mask			SBO			rotate			immediate													
Load/store immediate offset	cond [1]	0	1	0	P	U	B	W	L	Rn			Rd			immediate																
Load/store register offset	cond [1]	0	1	1	P	U	B	W	L	Rn			Rd			shift amount			shift	0	Rm											
Media instructions [4]: See Figure A3-2	cond [1]	0	1	1	x	x	x	x	x	x x x x x x x x x x x x x x x x													1	x x x x x								
Architecturally undefined	cond [1]	0	1	1	1	1	1	1	1	x x x x x x x x x x x x x x x x													1	1	1	1	x x x x x					
Load/store multiple	cond [1]	1	0	0	P	U	S	W	L	Rn			register list																			
Branch and branch with link	cond [1]	1	0	1	L	24-bit offset																										
Coprocessor load/store and double register transfers	cond [3]	1	1	0	P	U	N	W	L	Rn			CRd			cp_num			8-bit offset													
Coprocessor data processing	cond [3]	1	1	1	0	opcode1			CRn			CRd			cp_num			opcode2	0	CRm												
Coprocessor register transfers	cond [3]	1	1	1	0	opcode1			L	CRn			Rd			cp_num			opcode2	1	CRm											
Software interrupt	cond [1]	1	1	1	1	sw number																										
Unconditional instructions: See Figure A3-6	1	1	1	1	x x x x x x x x x x x x x x x x																											

Figure A3-1 ARM Instruction set summary

1. The cond field is not allowed to be 1111 in this line. Other lines deal with the cases where bits[31:28] of the instruction are 1111.
2. If the opcode field is of the form 10xx and the 5 field is 0, one of the following lines applies instead.
3. If the cond field is 1111, this instruction is UNPREDICTABLE prior to ARMv5.
4. The architecturally Undefined instruction uses a small number of these instruction encodings.

## Chapter A4 ARM Instructions

This chapter describes the syntax and usage of every ARM® instruction, in the sections:

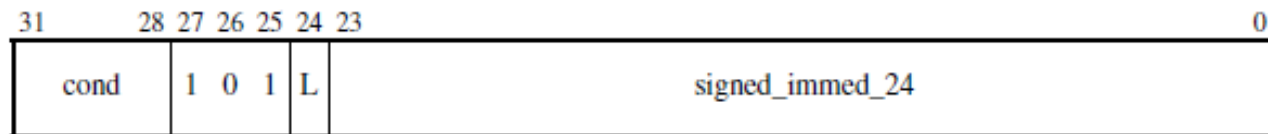
- *Alphabetical list of ARM instructions* on page A4-2
- *ARM instructions and architecture versions* on page A4-286.

# Example

## ◆ EA000006

- Instruction을 Binary로 변환
  - 1110 **1010** 0000 0000 0000 0000 0000 0110<sub>(2)</sub>
- 어떤 Instruction인지 Reference File을 통해 확인
  - B #6;
- Instruction이 어떤 의미를 가지는지 서술
  - PC+8+6x4 주소로 이동 (MIPS의 경우 +4이므로 주의)
    - PC는 현재 실행 중인 명령어의 주소를 의미함
    - 주소 단위가 4 Byte (= 1 Word)이기 때문임
  - 따라서,  $(0 + 8 + 24) / 4 = 008$ 번 주소로 이동
    - 다음 Instruction은 008번 주소에 있는 E59F2EC8이 됨

### A4.1.5 B, BL

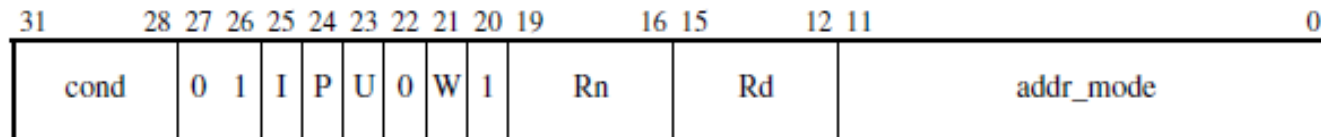


# Example

## ◆ E59FDE78

- Instruction을 Binary로 변환
  - 1110 0101 1001 1111 1101 1110 0111 1000<sub>(2)</sub>
- 어떤 Instruction인지 Reference File을 통해 확인
  - LDR \$13, [\$15, #0xE78];
  - 25번째 bit인 I bit가 0이므로, #으로 표기
- Instruction이 어떤 의미를 가지는지 서술
  - 15번 Register에 저장된 값에 #0xE78값을 더한 주소의 값을 메모리로 부터 읽어 와서 13번 Register에 저장
  - 메모리의 [\$15 + #0xE78] 주소에 저장된 값을 읽어와 \$13에 저장

### A4.1.23 LDR

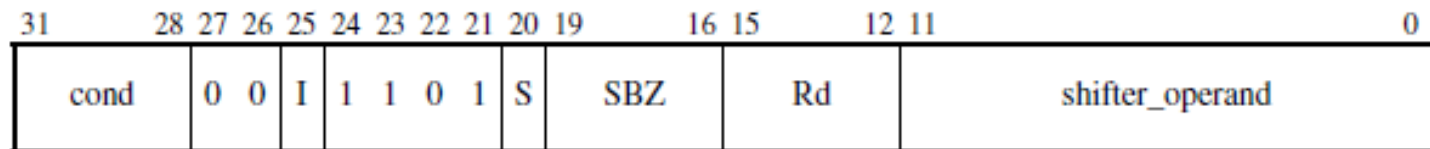


# Example

## ◆ E1A0F00E

- Instruction을 Binary로 변환
  - 1110 0001 101 0000 1111 0000 0000 1110<sub>(2)</sub>
- 어떤 Instruction인지 Reference File을 통해 확인
  - MOV \$15, #0xE;
- Instruction이 어떤 의미를 가지는지 서술
  - 15번 Register에 0xE (=14)를 저장

### A4.1.35 MOV





# Hint

- **Condition Code (e.g., 0 (0000) = Equal, E (1110) = Always)**
  - ARM Reference Manual ㄴ 112 Page
- **Data Processing Instructions (e.g., MOV, CMP)**
  - **Opcode:** Manual ㄴ 115 Page
  - **Instruction Encoding:** Manual ㄴ 116 Page
- **Load and Store Instructions (e.g., LDR, STR)**
  - **Address Mode:** Manual ㄴ 129 Page
  - **Instruction Encoding:** Manual ㄴ 130 Page
  - **Examples:** Manual ㄴ 131 Page
- **Branch Instructions (e.g., B, BL)**
  - **Instruction Encoding:** Manual ㄴ 160 Page
  - **Examples:** Manual ㄴ 114 Page

# 과제 채점 Focus

## ■ 과제 Report의 비중: 100%

- Instruction File (inst\_data.mif) 내 000번 ~ 024번 주소의 Instruction 분석
- Report에 반드시 포함되어야 하는 내용
  - 학과, 학번, 이름, 제출 일자
  - 각 Instruction에 대해
    - 16진수 명령어를 2진수로 변환 후 (1점)  
어떤 Instruction인지 Reference 파일을 통해 확인 후  
Instruction이 어떤 의미를 가지는지 서술 (2점)
  - 분석한 의미에 맞게 실제 동작 순서 기술 (1점)
  - 동작이 어디서 끝나는지 명시 (끝나지 않는다면, 어디부터 어디까지 반복된다고 구체적으로 명시) (1점)

# 과제 채점 Focus

## ■ 제출 기한

- 4월 16일 (금) 11:59 PM

## ■ 제출 방법

- 스마트 캠퍼스를 통해 파일들 제출
- 제출할 파일 목록
  - 보고서
- 파일 제출 방법
  - "ca1\_학번.zip"으로 압축하여 제출