# ARM Architecture (1A)

Young Won Lim
3/9/18

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice.

Young Won Lim
3/9/18

# Based on

ARM System-on-Chip Architecture, 2nd ed, Steve Furber

# Architectural Inheritance

A load-store architecture

Fixed-length 32-bit instructions

3-address instruction formats

# ARM Instruction Set

The load-store architecture

3-address data processing instructions

(2 source registers + 1 destination register)

Conditionally executes every instruction

Multiple data transfer instruction

Single cycle execution of shift and ALU operations

Open instruction set for coprocessors

A very dense 16-bit compressed instruction set (Thumb)

# ARM Exceptions

Interrupts

Traps

Supervisor calls

# ARM Exception Types

- Reset
- Non-maskable Interrupt
- Hard Fault
- Memory Management
- Bus Fault
- Usage Fault
- SVCall
- Debug Monitor
- PendSV
- Sys Tick
- External Interrupt

# ARM Processor Operating Modes

- User – normal mode for most programs (tasks)
- FIQ (Fast Interrupt)
- IRQ (Interrupt)
- SVC (Supervisor)
- ABT (Abort)
- UND (Undefined)
- System – use the same registers as User mode

# ARM Processor Regular Registers

| R0 |
|---|
| R1 |
| R2 |
| R3 |
| R4 |
| R5 |
| R6 |
| R7 |
| R8 |
| R9 |
| R10 |
| R11 |
| R12 |
| R13 (SP) |
| R14 (LR) |
| R15 (PC) |

| CPSR |
|---|

# Each mode accesses a subset of regular registers

| User | System | Fast Interrupt | Interrupt | Supervisor | Abort | Undefined |
|---|---|---|---|---|---|---|
| R0 | R0 | R0 | R0 | R0 | R0 | R0 |
| R1 | R1 | R1 | R1 | R1 | R1 | R1 |
| R2 | R2 | R2 | R2 | R2 | R2 | R2 |
| R3 | R3 | R3 | R3 | R3 | R3 | R3 |
| R4 | R4 | R4 | R4 | R4 | R4 | R4 |
| R5 | R5 | R5 | R5 | R5 | R5 | R5 |
| R6 | R6 | R6 | R6 | R6 | R6 | R6 |
| R7 | R7 | R7 | R7 | R7 | R7 | R7 |
| R8 | R8 | | R8 | R8 | R8 | R8 |
| R9 | R9 | | R9 | R9 | R9 | R9 |
| R10 | R10 | | R10 | R10 | R10 | R10 |
| R11 | R11 | | R11 | R11 | R11 | R11 |
| R12 | R12 | | R12 | R12 | R12 | R12 |
| R13 (SP) | R13 (SP) | | | | | |
| R14 (LR) | R14 (LR) | | | | | |
| R15 (PC) | R15 (PC) | R15 (PC) | R15 (PC) | R15 (PC) | R15 (PC) | R15 (PC) |
| CPSR | CPSR | CPSR | CPSR | CPSR | CPSR | CPSR |

# ARM Processor Special Registers

## additional hardware registers

| Registers for **Fast Interrupt** | Registers for **Interrupt** | Registers for **Supervisor** | Registers for **Abort** | Registers for **Undefined** |
|---|---|---|---|---|
| R8_fig | | | | |
| R9_fig | | | | |
| R10_fig | | | | |
| R11_fig | | | | |
| R12_fig | | | | |
| R13_fig | R13_irq | R13_svc | R13_abt | R13_und |
| R14_fig | R14_irq | R14_svc | R14_abt | R14_und |
| SPSR_fiq | SPSR_irq | SPSR_svc | SPSR_abt | SPSR_und |

# ARM Processor Registers

| User | System | Fast Interrupt | Interrupt | Supervisor | Abort | Undefined |
|---|---|---|---|---|---|---|
| R0 | R0 | R0 | R0 | R0 | R0 | R0 |
| R1 | R1 | R1 | R1 | R1 | R1 | R1 |
| R2 | R2 | R2 | R2 | R2 | R2 | R2 |
| R3 | R3 | R3 | R3 | R3 | R3 | R3 |
| R4 | R4 | R4 | R4 | R4 | R4 | R4 |
| R5 | R5 | R5 | R5 | R5 | R5 | R5 |
| R6 | R6 | R6 | R6 | R6 | R6 | R6 |
| R7 | R7 | R7 | R7 | R7 | R7 | R7 |
| R8 | R8 | R8_fig | R8 | R8 | R8 | R8 |
| R9 | R9 | R9_fig | R9 | R9 | R9 | R9 |
| R10 | R10 | R10_fig | R10 | R10 | R10 | R10 |
| R11 | R11 | R11_fig | R11 | R11 | R11 | R11 |
| R12 | R12 | R12_fig | R12 | R12 | R12 | R12 |
| R13 (SP) | R13 (SP) | R13_fig | R13_irq | R13_svc | R13_abt | R13_und |
| R14 (LR) | R14 (LR) | R14_fig | R14_irq | R14_svc | R14_abt | R14_und |
| R15 (PC) | R15 (PC) | R15 (PC) | R15 (PC) | R15 (PC) | R15 (PC) | R15 (PC) |
| CPSR | CPSR | CPSR | CPSR | CPSR | CPSR | CPSR |
| | | SPSR_fiq | SPSR_irq | SPSR_svc | SPSR_abt | SPSR_und |

http://www.cs.otago.ac.nz/cosc440/readings/arm-syscall.pdf

# ARM Exception Handling

1. Save the current state
   copying the **PC** into **r14**_exec
   copying the **CPSR** into **SPSR**_exec
   exec : exception type
2. changing the processor operating mode
   To the appropriate exception mode
3. **PC** is forced to have values between **0x00** and **0x1C**
   depending on the type of exception
   exception handlers

## References

[1] ftp://ftp.geoinfo.tuwien.ac.at/navratil/HaskellTutorial.pdf
[2] https://www.umiacs.umd.edu/~hal/docs/daume02yaht.pdf