

---

# ARM Assembly Programming

Dae Hyun Kim

EECS  
Washington State University

---

# Register File

- 16 registers
  - R0 ~ R12: General purpose registers
  - R13: Stack Pointer
  - R14: Link Register
  - R15: Program Counter

|                       |
|-----------------------|
| R15 (Program Counter) |
| R14 (Link Register)   |
| R13 (Stack Pointer)   |
| R12                   |
| R11                   |
| R10                   |
| R9                    |
| R8                    |
| R7                    |
| R6                    |
| R5                    |
| R4                    |
| R3                    |
| R2                    |
| R1                    |
| R0                    |

# Instructions

---

- In general, an instruction looks like this:
  - instruction Rd, Rs1, Rs2
    - Rd: Destination register
    - Rs1: Source register 1
    - Rs2: Source register 2

# Instructions

---

- MOV Rd, Rs
    - Example: MOV R1, R2
    - Copy the content of R2 to R1
  
  - MOV Rd, #const
    - Example: MOV R3, #0x34
    - Load R3 with constant 0x34 (i.e., R3 = 0x34)
    - This (kind of) instruction is called “MOV immediate”, where “immediate” means a constant.
    - The constant should be an 8-bit value.
    - The constant could be a decimal number like #45 or a hexadecimal number like #0x46 (which is 70).
    - The rest of the bits will be 0.
    - MOV R3, #0xF3 => R3: 0000 0000 0000 0000 0000 0000 1111 0011
-

# Instructions

---

- LDR Rd, =const
    - The constant could be a 32-bit value.
    - This instruction is used to load a half-word (16-bit) or word (32-bit) constant value into a register.
    - (Notice that MOV Rd, const is used to load a byte (8-bit) value into a register.)
    - LDR R2, =0x41210000
      - R2 becomes 0100 0001 0010 0001 0000 0000 0000 0000
-

# Instructions

---

- ADD Rd, Rs1, Rs2
    - Example: ADD R1, R2, R3 ( $R1 = R2 + R3$ )
  - ADD Rd, Rs, const
    - Example: ADD R1, R2, #0x12 ( $R1 = R2 + 18$ )
  - SUB Rd, Rs1, Rs2
    - Example: SUB R1, R2, R3 ( $R1 = R2 - R3$ )
  - SUB Rd, Rs, const
    - Example: SUB R1, R2, #0x12
  - AND (bitwise AND)
  - ORR (bitwise OR)
  - EOR (bitwise exclusive OR)
  - MVN Rd, Rs ( $Rd =$  bitwise inversion of Rs)
  - MVN Rd, const
-

# Instructions

---

- LDR Rd, [Rs]
    - Example: LDR R1, [R3]
    - Load the content stored at the memory address [R3] into R1.
  - STR Rd, [Rs]
    - Example: STR R2, [R4]
    - Store the content of R2 to the memory address [R4].
  - B address\_label
    - Jump to the address label (Unconditional jump)
-