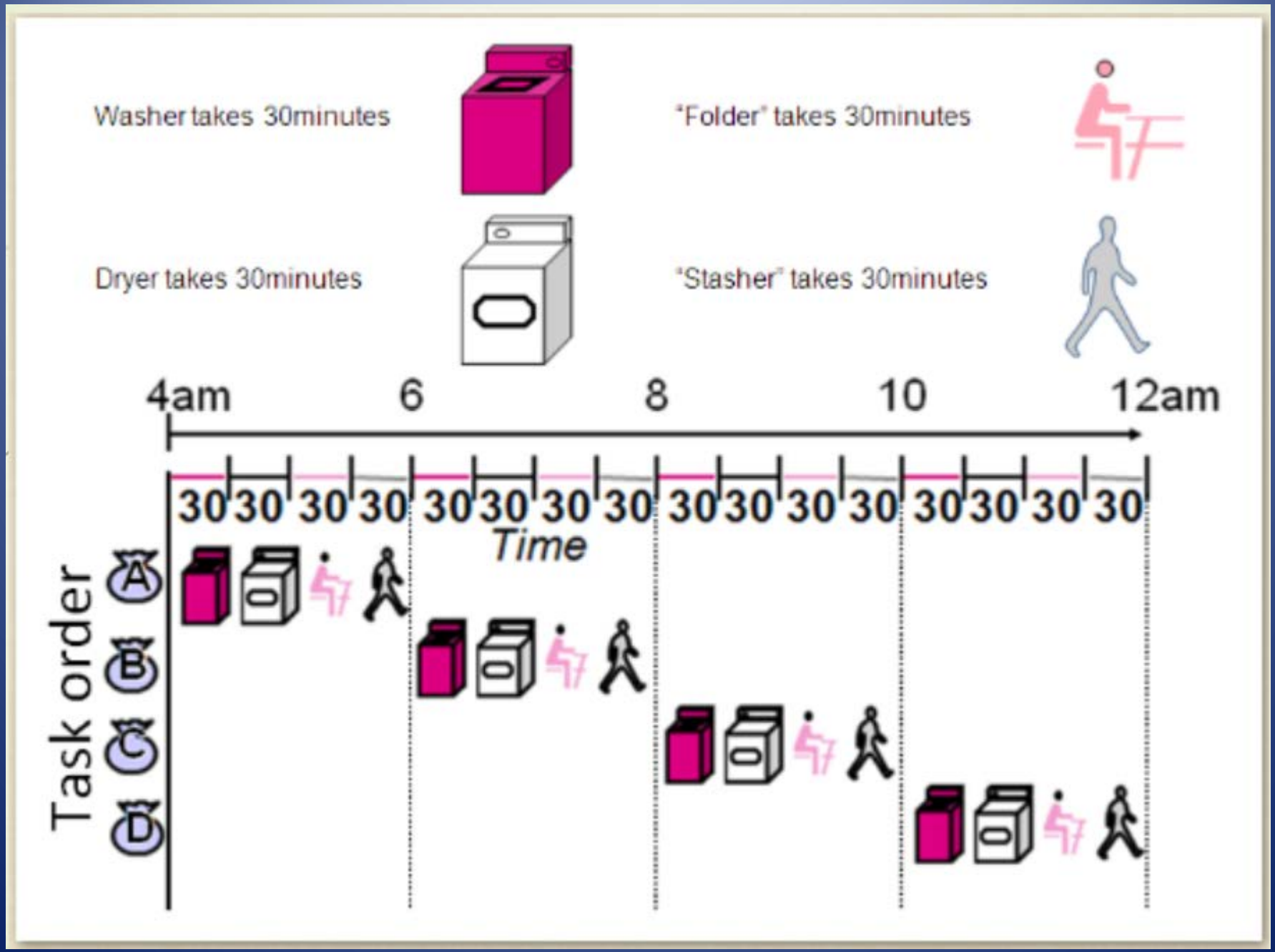
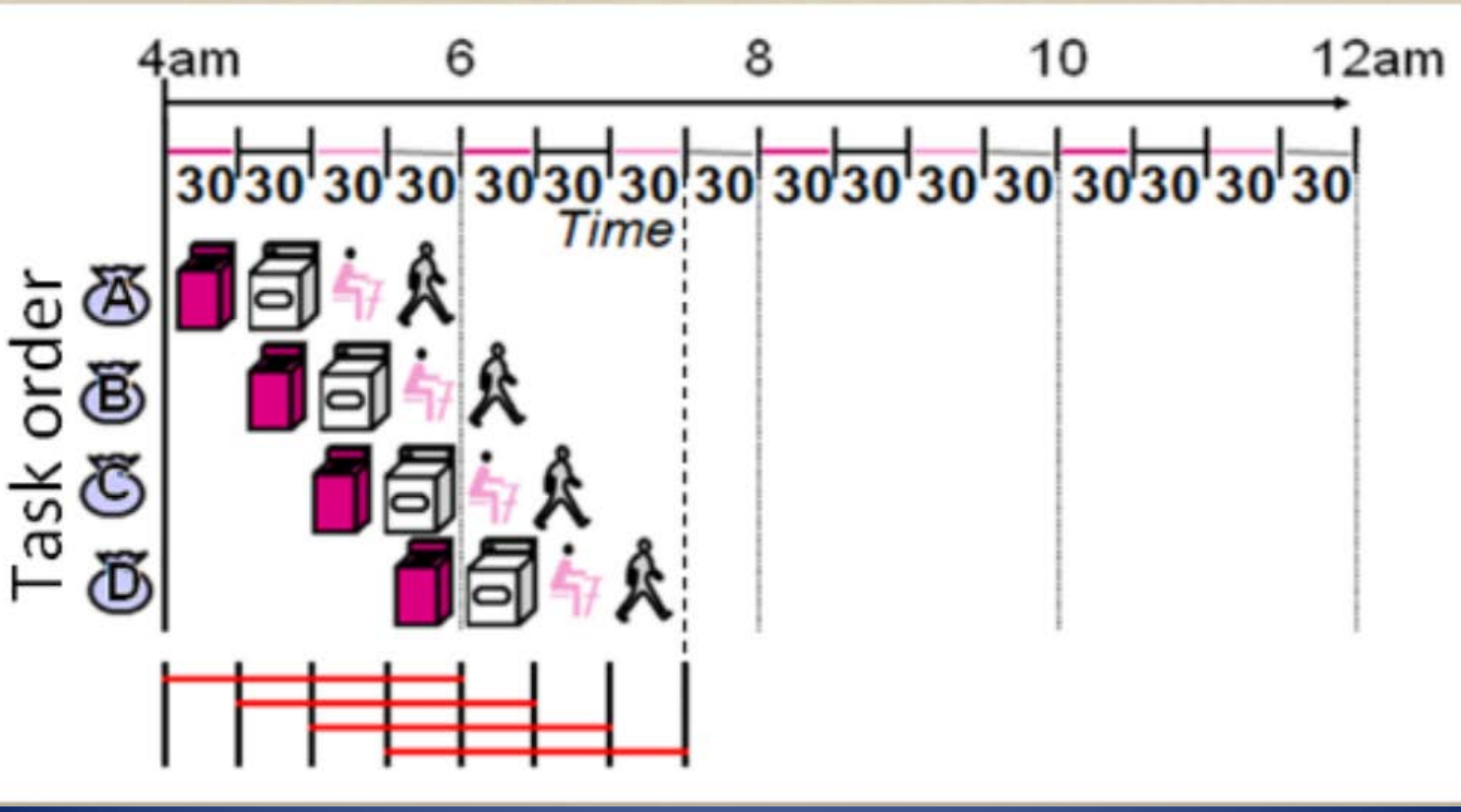
Pseudocode for basic steps of execution cycle

```
LOOP    Fetch next instruction
        Advance PC
        Decode the instruction
        Execute the instruction
        Continue loop
```


Washer takes 30minutes
"Folder" takes 30minutes
Dryer takes 30minutes
"Stasher" takes 30minutes
4am
6
8
10
12am
30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30
Time
Task order
A
B
C
D


4am
6
8
10
12am
30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30
Time
Task order
A
B
C
D

Pipeline

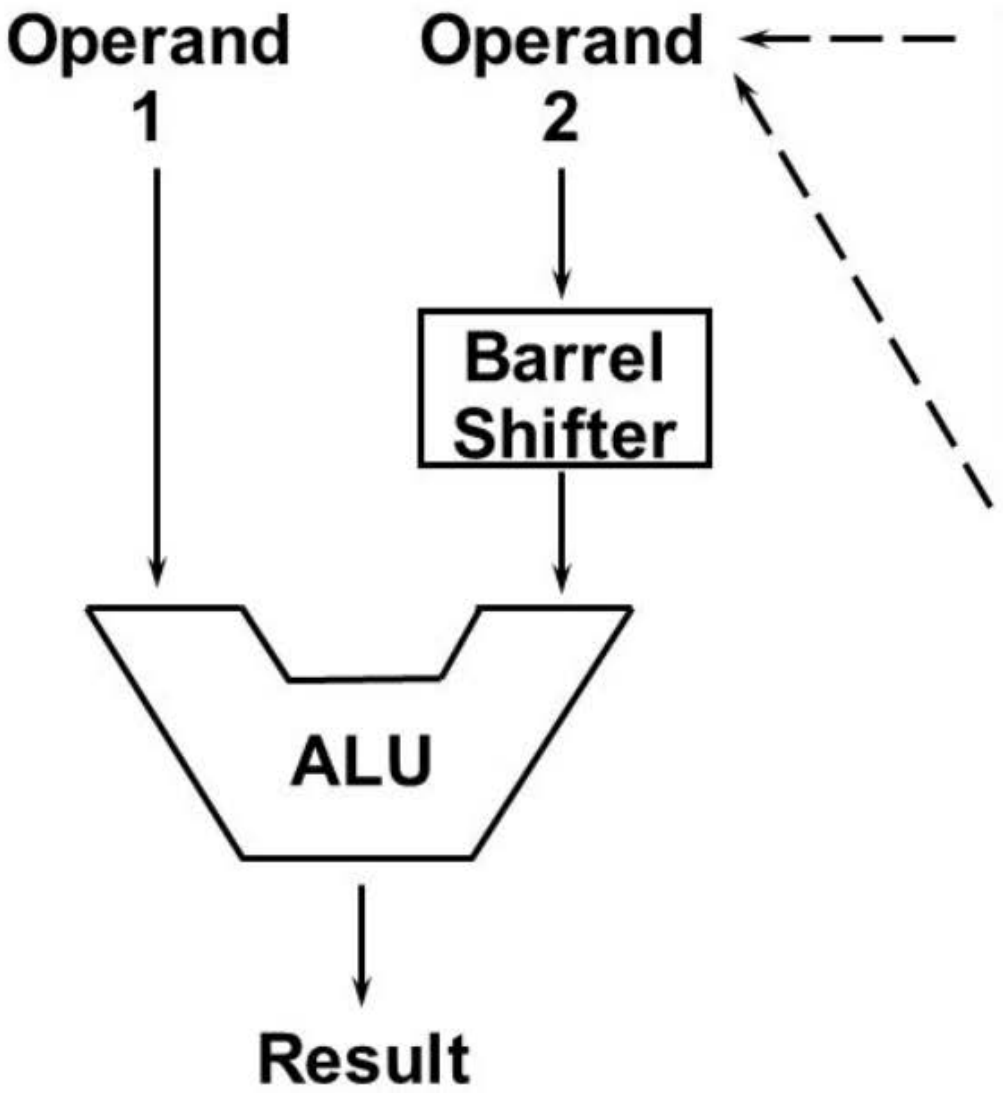
- In theory, each instruction is one instruction cycle
- In practice, there is interdependency between instructions
  - Solution: instruction scheduling

Branch instructions flush and refill the instruction pipeline.

The ARM Barrel Shifter

- ARM architectures have a unique piece of hardware known as a barrel shifter.
  - Device moves bits in a word left or right.
- Most processors have stand alone instructions for shifting bits.
- ARM allows shifts as part of regular instructions.
- Allows for quick multiplication and division.

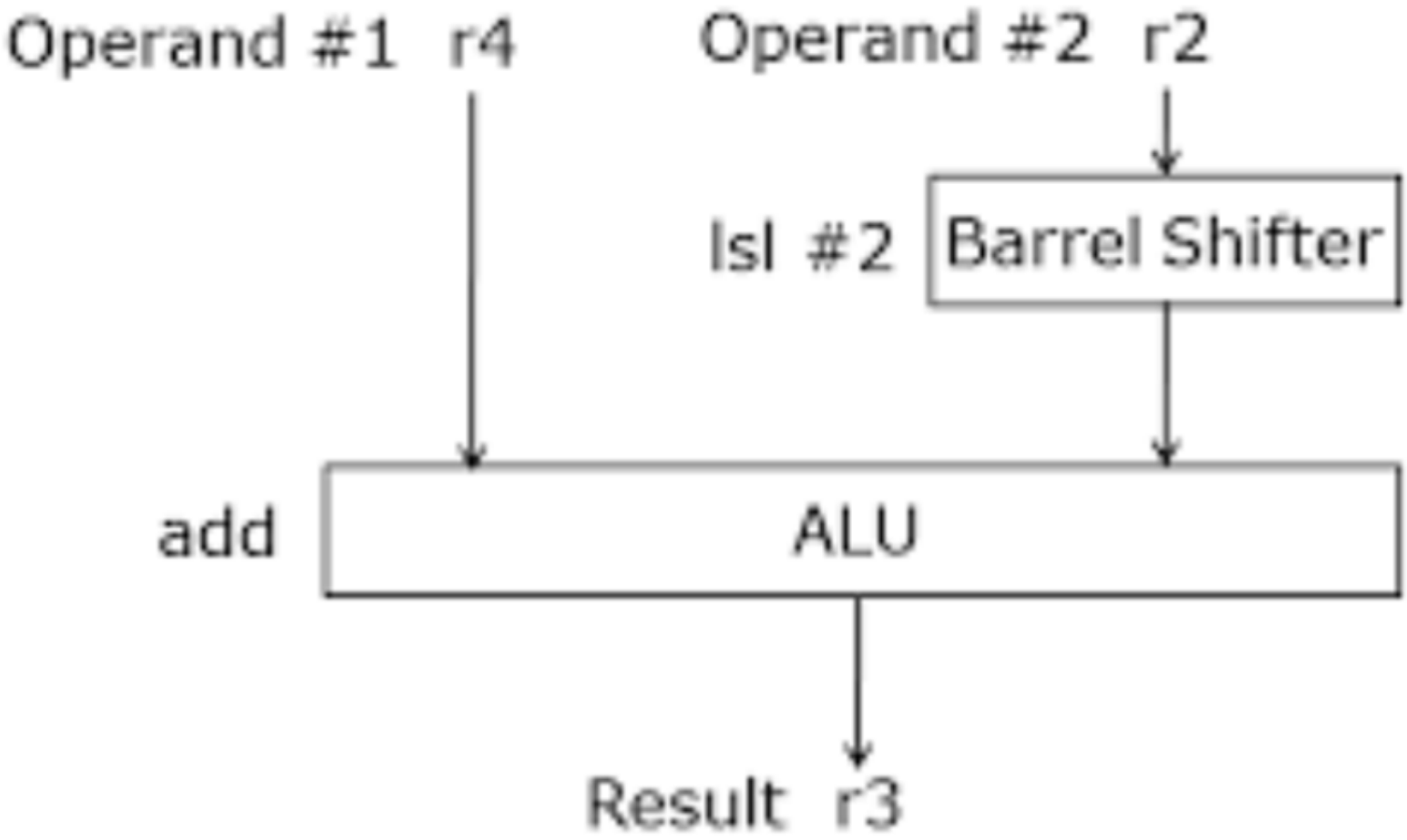
Using the Barrel Shifter: The Second Operand



* **Register, optionally with shift operation applied.**
* **Shift value can be either be:**
  - 5 bit unsigned integer
  - Specified in bottom byte of another register.

* **Immediate value**
  - 8 bit number
  - Can be rotated right through an even number of positions.
  - Assembler will calculate rotate for you from constant.

add r3, r4, r2, lsl #2; r3 = r4 + (r2 << 2)

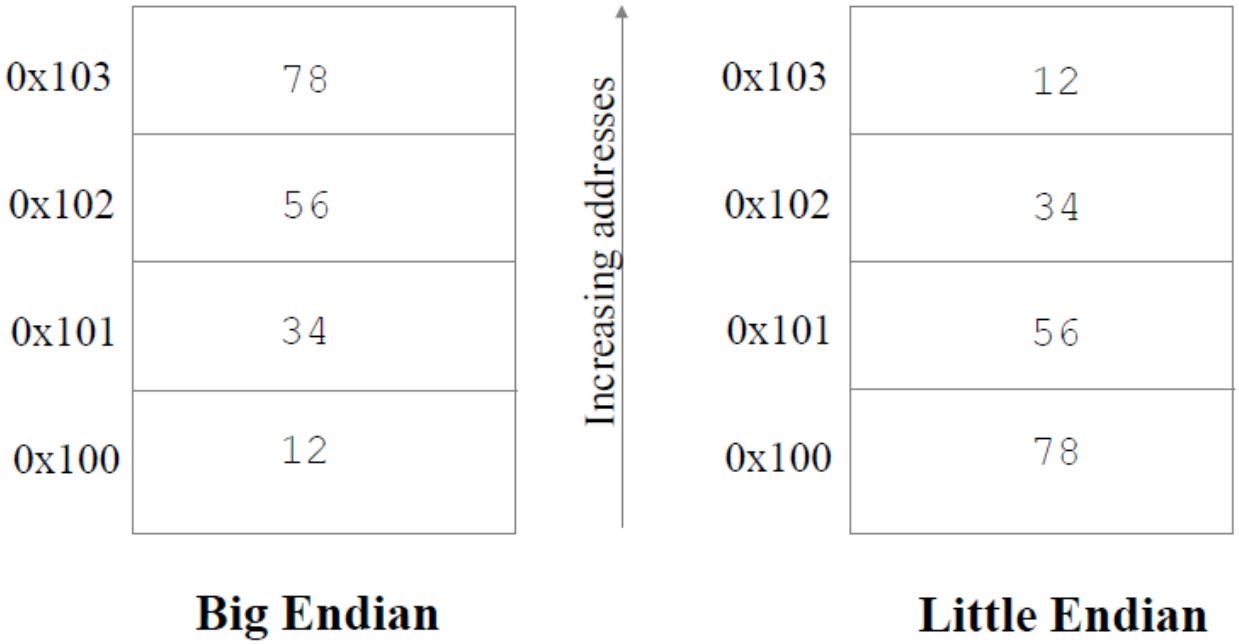


ARM Data Sizes and Instructions

- The ARM is a 32-bit RISC architecture, so in relation to that:
  - **Byte** means 8 bits
  - **Halfword** means 16 bits (two bytes)
  - **Word** means 32 bits (four bytes)
- ARM cores can be configured to view words stored in memory as either Big-Endian or Little-Endian format.

Big Endian vs. Little Endian

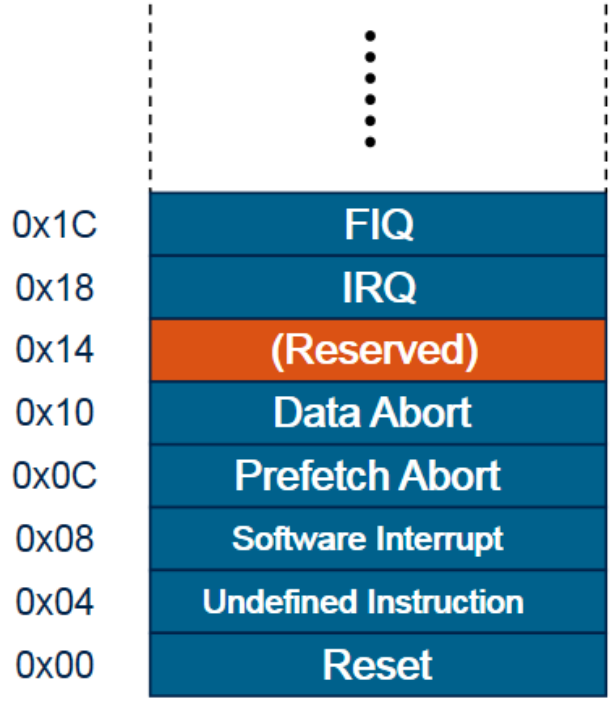
How 0x12345678 would be stored in a 32-bit memory?



Exception Handling

- **When an exception occurs, the ARM:**
  - Copies CPSR into SPSR_<mode>
  - Sets appropriate CPSR bits
    - Change to ARM state
    - Change to exception mode
    - Disable interrupts (if appropriate)
  - Stores the return address in LR_<mode>
  - Sets PC to vector address
- **To return, exception handler needs to:**
  - Restore CPSR from SPSR_<mode>
  - Restore PC from LR_<mode>

This can only be done in ARM state.



Vector Table

Vector table can be at `0xFFFF0000` on ARM720T and on ARM9/10 family devices