

SYS-102 Basic Computer Architecture
Assessment Activity SYS-102:00070
Instruction Set Architecture

Name: (Win) Thanawin Pattanaphol

Student ID: 01324096

CPU Architecture and Registers

Register	Register Full Name	Explanation
PC	Program Counter	Holds the address of the following instruction to execute.
A	Accumulator	General purpose for holding temporary data that is being processed.
MAR	Memory Address Register	Holds the address for accessing in memory
MDR	Memory Data Register	Holds data read from or written to memory.
CR	Checksum Register	Register for storing checksum during calculation and verification.
IP	Instruction Pointer	Pointer to the current instruction being executed.

Memory Layout

0x00 – 0x0F (Instructions – 16 bits)	0x10 – 0x15: 6-bit message	0x16 (checksum)
--------------------------------------	----------------------------	-----------------

Instruction Set Architecture (ISA)

Instruction Format:

- Opcode (4 bits) – Determines the operation
- Operands (12 bits) – Specifies register and/or memory address involved in the operation.

Instruction	Opcode	Operands	Explanation
LD	0b0001	A, address	Loads data from address to A
STR	0b0010	A, address	Store value from register A to address
MOV	0b0011	A, B	Move data from B to A
ADD	0b0100	A, B	Add content of B and A and store result in A
XOR	0b0101	A, B	Performs XOR operation between A and B and store result in A
JMP	0b0110	address	Jumps to specified memory address.
HLT	0b0111	none	Halts the CPU
CMP	0b1000	A, checksum	Compares the value in register A with checksum value in memory.

SYS-102 Basic Computer Architecture
Assessment Activity SYS-102:00070
Instruction Set Architecture

Name: (Win) Thanawin Pattanaphol

Student ID: 01324096

Message Verification

Here is an example program that process a 6-character message, one character by one, calculate a checksum and compare it with the given checksum.

```
; Checksum register
LD CC, 0x00           ; Set checksum to 0
LD MAR, 0x10         ; Set start address for message

; Loop through the 6 characters
LOOP:
    LD MAR, A         ; Load current character into A
    ADD CC, A         ; Add value of char into checksum
    MOV CC, A         ; Store updated checksum in CC
    ADD MAR, 1        ; Move to next memory address
    CMP MAR, 0x16     ; Check if we've proceeded all chars
    JMP LOOP          ; Load if not finished

; Load transmitted checksum
LD A, 0x16           ; Load transmitted checksum
CMP CC, A            ; Compare computed checksum with the
                        ; transmitted one
JMP MATCH            ; Jump if match

ERROR:
    JUMP ERROR        ; Goes into infiinite loop

MATCH:
    HALT              ; Stop the CPU if successful
```

Explanation

- First Section: Setting the value of register the checksum register to zero (default value) and set the start address for the message to the first character (0x10)
- Second Section: Looping through all the characters. In each iteration, we add the value of the character to the checksum register, move to the next character and add the next character's value to the checksum register and repeat.
- Third Section: Comparing if the calculated checkup and the transmitted checksum in the 0x16 bit is equal or not. If so, jump to the match section which halts the CPU, if not, jump to the error section that results in an infinite loop.