

7-2

CALL 1110 => SP := SP - 1; M[SP] := PC; PC := X

RETV 1111 ~~1000~~ 0- => PC := M[SP]; SP := SP + 1

PUSH 1111 ~~1010~~ 0100 0- => SP := SP - 1; M[SP] := AC

POP 1111 0110 0- => AC := M[SP]; SP := SP + 1

PUSHI 1111 0000 0- => SP := SP - 1; M[SP] := M[AC]

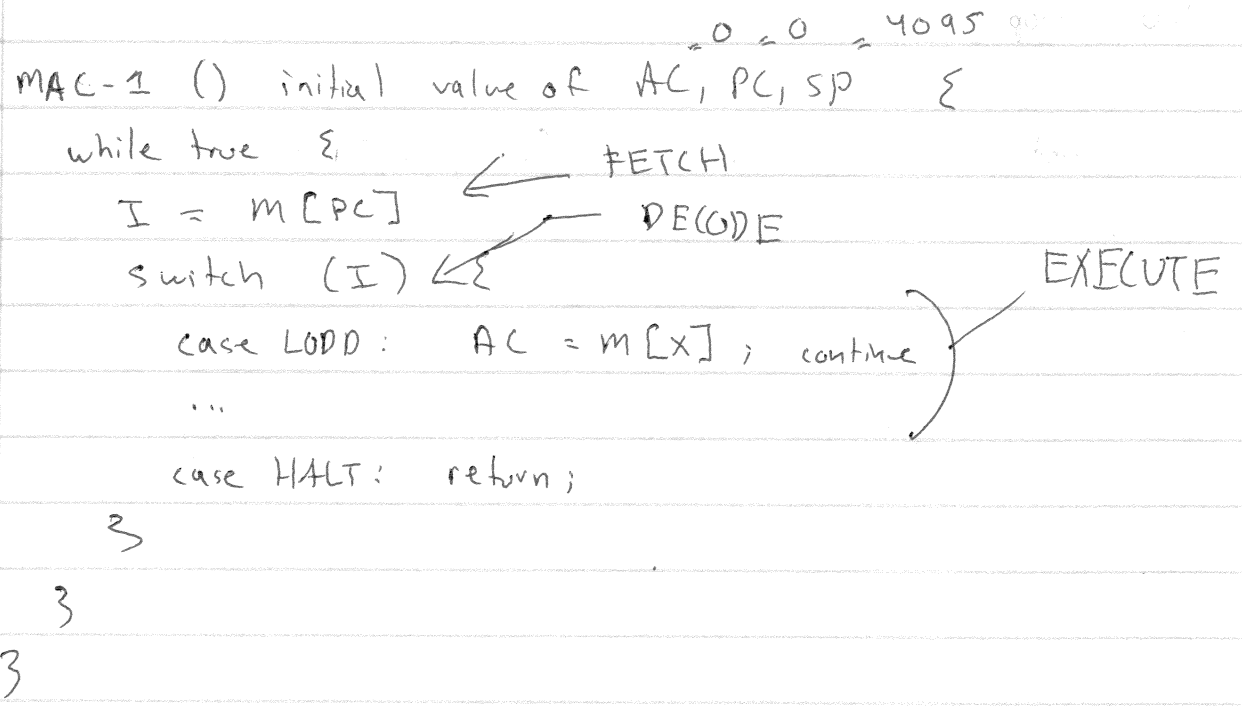
POPI 1111 0010 0- => M[AC] := M[SP]; SP := SP + 1

SWAP 1111 1010 0- => swap AC and SP

INSP 1111 1100 8y => SP := SP + y (0 ≤ y ≤ 255)

DESP 1111 1110 8y => SP := SP - y

HALT 1111 1111 0- => stops machine



Assume that "A" is a location and "B" is a location and "N" is a location
↙ ptr to A DATA ↙ ptr to B DATA ↘ number
↓ ↓
N values N values

Purpose: Push $A[i] + B[i]$ onto the stack for $i \in [0, N]$
Leave AC as the total of all A & B

Purpose: Compute all the Fibonacci numbers the MIC-1 can
↳ Save them to the stack

```
STOL 0 // AC = 0 => M[SP] = Fib(0) = 0
LOCO 1 //          => AC = 1
STOL 1 // AC = 1 => M[SP+1] = Fib(1) = 1
loop: ADDL 0 // AC = Fib(n+1) M[SP] = Fib(n) => AC = Fib(n+2)
      STOL 2 //          M[SP+2] = Fib(n+2)
      INSP 1 // AC = Fib(n+1) M[SP] = Fib(n)
                        ↑      n' = n+1      ↓
      JPOS loop:
```

Instruction encoding RISC / MIPS