

Single Cycle Processor

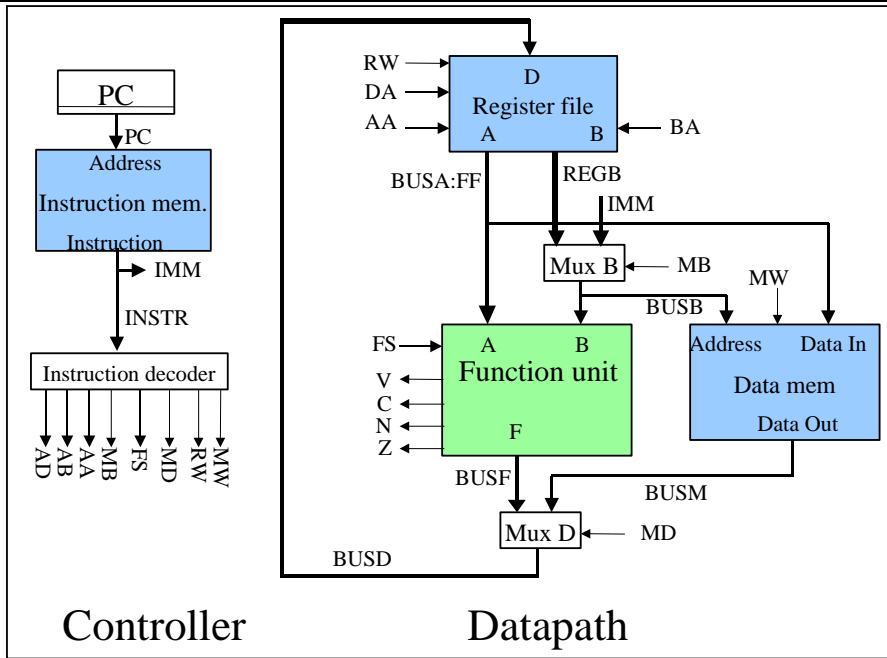


Figure 1: single cycle processor

MW: 1 \Rightarrow write memory, 0 \Rightarrow no write
 MD: 1 \Rightarrow Mem, 0 \Rightarrow F
 RW: 1 \Rightarrow write in reg. File, 0 \Rightarrow no write
 MB: 1 \Rightarrow IMM, 0 \Rightarrow regB
 AA, BA, DA address selection reg. file R0 - R15

FS	function
00000	F = A
00001	F = A + 1
00010	F = A + B
00011	F = A + B + 1
00100	F = A + Bn
00101	F = A + Bn + 1
00110	F = A - 1
00111	F = B
01000	F = A \wedge B
01010	F = A \vee B
01100	F = A \oplus B
01110	F = A n
10000	F = sr A (shift right)
10001	F = sl A (shift left)

Pipelined processor

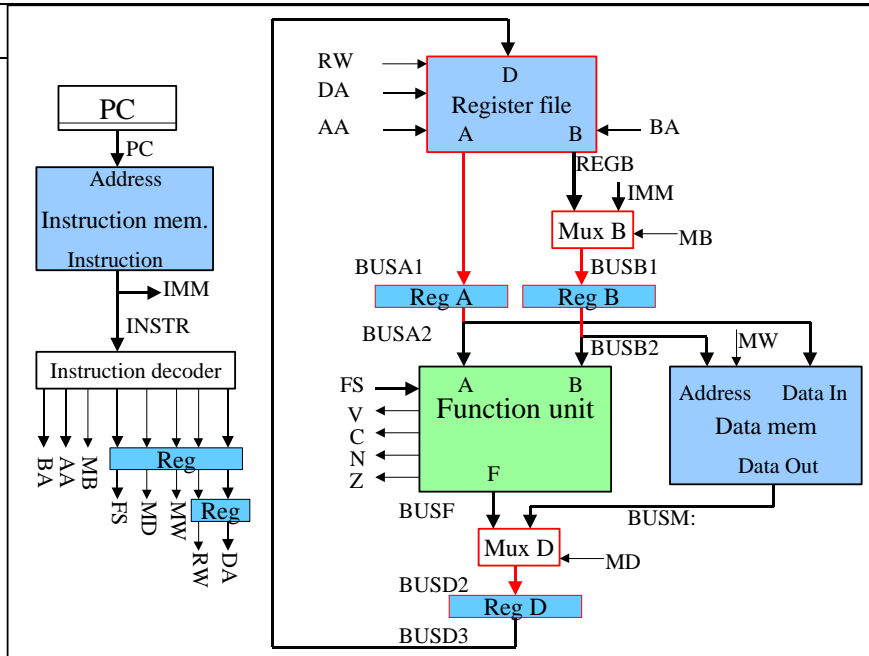


Figure 2: pipelined processor

Multicycle Processor

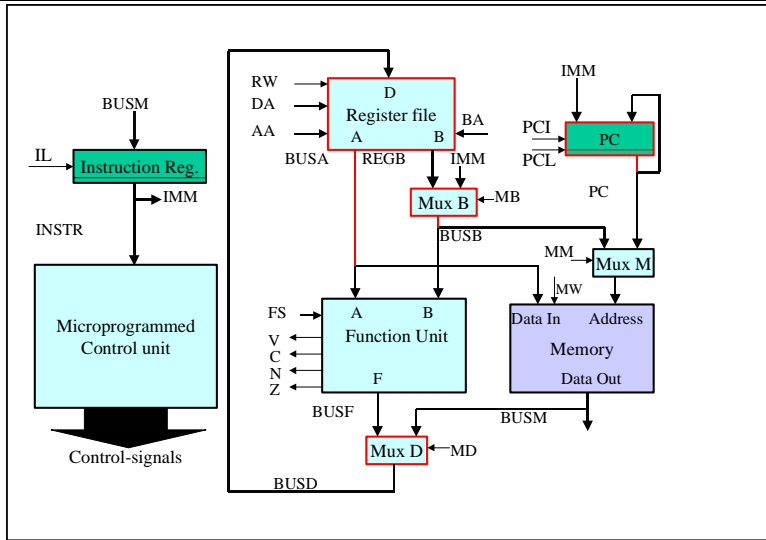


Figure 3:
multi cycle processor

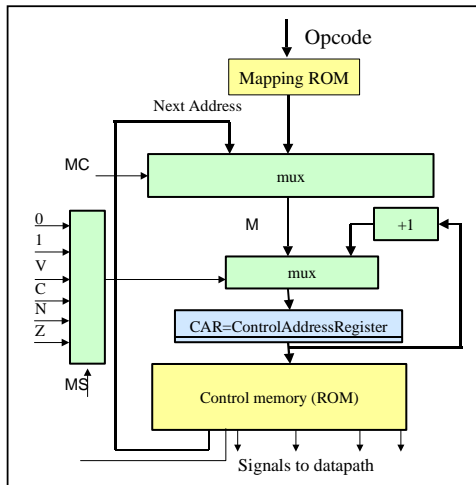


Figure 4:
sequencer for
multi cycle processor

FS	function
00000	F = A
00001	F = A + 1
00010	F = A + B
00011	F = A + B + 1
00100	F = A + Bn
00101	F = A + Bn + 1
00110	F = A - 1
00111	F = B
01000	F = A ^ B
01010	F = A v B
01100	F = A ⊕ B
01110	F = An
10000	F = sr A (shift right)
10001	F = sl A (shift left)

- MC: 1 ⇒ Opcode (via Mapping Rom), 0 ⇒ Next Address
 MS: bitpat symbolic (~C and ~Z are not available in ProcSim)
 000 0 CAR ≤ M
 001 1 CAR ≤ CAR + 1
 010 C if C=1 then CAR ≤ CAR + 1 else CAR ≤ M
 011 V if V=1 then CAR ≤ CAR + 1 else CAR ≤ M
 100 Z if Z=1 then CAR ≤ CAR + 1 else CAR ≤ M
 101 N if N=1 then CAR ≤ CAR + 1 else CAR ≤ M
 110 ~C if C=0 then CAR ≤ CAR + 1 else CAR ≤ M
 111 ~Z if Z=0 then CAR ≤ CAR + 1 else CAR ≤ M
 MM: 1 ⇒ PC, 0 ⇒ busB
 IL: 1 ⇒ Write Instruction register, 0 ⇒ no write

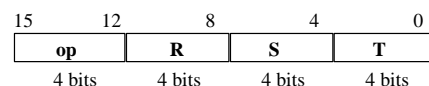
- PCI: 1 ⇒ increment PC
 PCL: 1 ⇒ Load PC with IMM
 DA, BA, AA address selection reg. file R0 - R15
 MB: 0 ⇒ regB; 1 ⇒ IMM
 MD: 1 ⇒ Mem, 0 ⇒ F
 RW: 1 ⇒ write in reg. file, 0 ⇒ no write
 MW: 1 ⇒ write memory, 0 ⇒ no write

Note: If PCI and PCL are both '1' the value of the program counter is undefined. If PCI and PCL are both '0' the value of the program counter is unchanged.

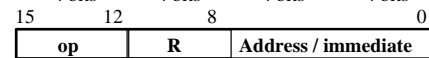
General information for all three processors

Program Counter and each register in the register file is 8 bits
 Instruction Register 16 bits
 Memory contains 256 addresses, each address contains 1 byte
 All instructions are 16 bits long
 Three instruction formats:

R-type



I-type



J-type

