Instruction Set Being Considered:

- Load and Store Instructions

```
LW   R2,124(R8)     // R2 = Mem[R8+124]
SW   R5,-10(R25)    // Mem[R25-10] = R5
```

- Arithmetic and Logic Instructions (only register operands)

```
ADD   R1,R2,R3      // R1 = R2 + R3
ADD   R1,R2,R0      // R1 = R2 + 0
SUB   R12,R10,R8    // R12 = R10 - R8
AND   R20,R1,R5     // R20 = R1 & R5
OR    R11,R5,R6     // R11 = R5 | R6
MUL   R5,R6,R7      // R5 = R6 * R7
SLT   R5,R11,R12    // If R11 < R12, R5=1; else R5=0
```

- Arithmetic and Logic Instructions (immediate operand)

```
ADDI   R1,R2,25     // R1 = R2 + 25
SUBI   R5,R1,150    // R5 = R1 - 150
SLTI   R2,R10,10    // If R10<10, R2=1; else R2=0
```

- Branch Instructions

```
BEQZ   R1,Loop      // Branch to Loop if R1=0
BNEQZ  R5,Label     // Branch to Label if R5!=0
```

- Jump Instruction

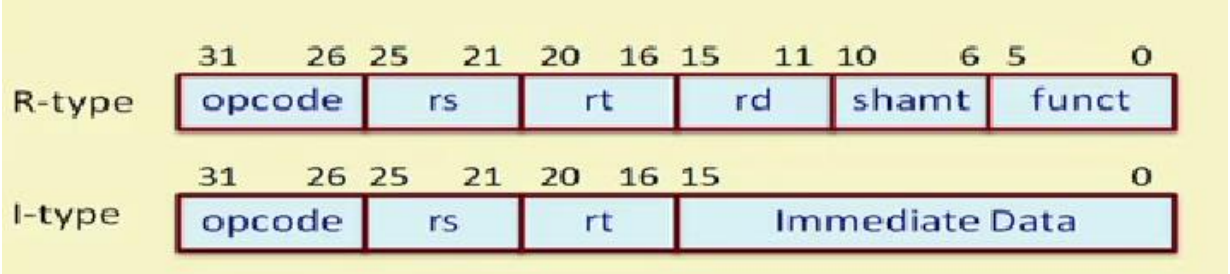
```
J      Loop         // Branch to Loop unconditionally
```

- Miscellaneous Instruction

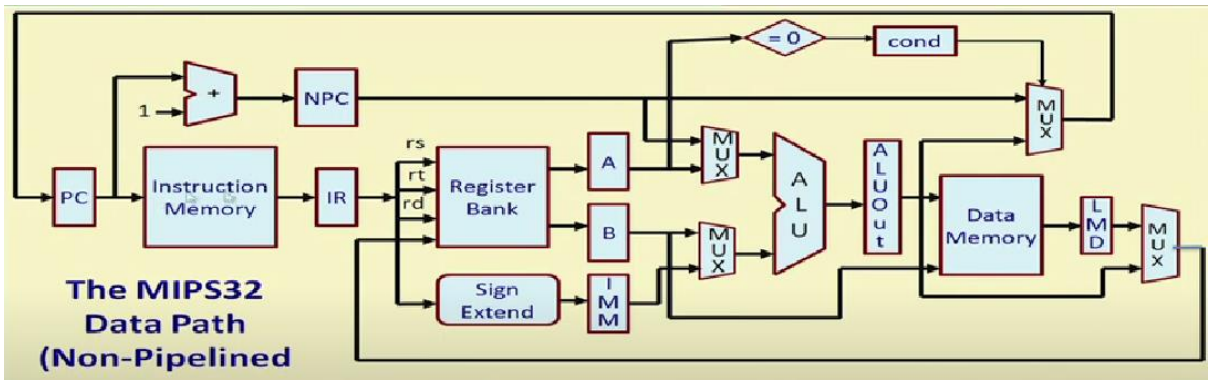
```
HLT                 // Halt execution
```

Instructions with their Opcodes:

Instruction	opcode	Instruction	opcode
ADD	000000	LW	001000
SUB	000001	SW	001001
AND	000010	ADDI	001010
OR	000011	SUBI	001011
SLT	000100	SLTI	001100
MUL	000101	BNEQZ	001101
HLT	111111	BEQZ	001110



MIPS32 Datapath without Pipelining:



MIPS32 Datapath with Pipelining: