Pipeline Implementation of MIPS32

It is a pipeline implementation of MIPS32 which is basically a Reduced Instruction Set Computer (RISC) architecture. It will have a small subset of the instructions (and some simplifying assumptions).

Some features that our MIPS32 will have are,
> MIPS32 registers:
  a) 32, 32-bit general-purpose registers (GPRs), R0 to R31.
      * Register R0 contains a constant 0, can't be written.
  b) A special-purpose 32-bit program counter (PC).
      * Points to the next instruction in memory to be fetched and executed.
> NO Flag register (zero, carry, sign, etc...)
> Very few addressing modes (register, immediate, register indexed, etc...)
      * Only load and store instructions can access memory.
> We assume memory word size is 32 bits (word addressable).

Instructions Subset of our MIPS32:
> Load and Store Instruction:

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

>Arithmetic and Logic Instruction (only register operands)

```
ADD R1,R2,R3          //R1 = R2 + R3
ADD R1,R2,R0          //R1 = R2 + 0  (similar to MOV instruction)
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 Instruction (immediate operands)

```
ADDI R1,R2,25         //R1 = R2 + 25
SUBI R15,R1,35        //R15 = R1 - 35
SLTI R7,R14,10        //If R14 < 10, R7=1; else R7=0
```

>Branch Instructions

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

>Jump Instruction

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

>Miscellaneous Instruction

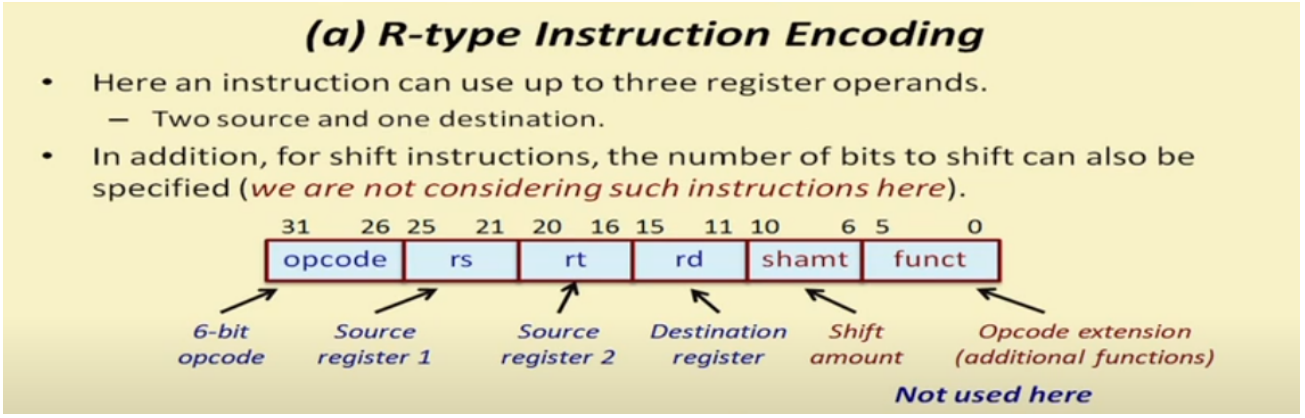
HLT //Halt execution

>All MIPS32 instructions can be classified into three groups in terms of instruction encoding

*R-type (Register), I-type (Immediate), and J-type (Jump)

*In an instruction encoding, the 32 bits of the instruction are divided into several fields of fixed width.

*All instructions may not use all the fields.

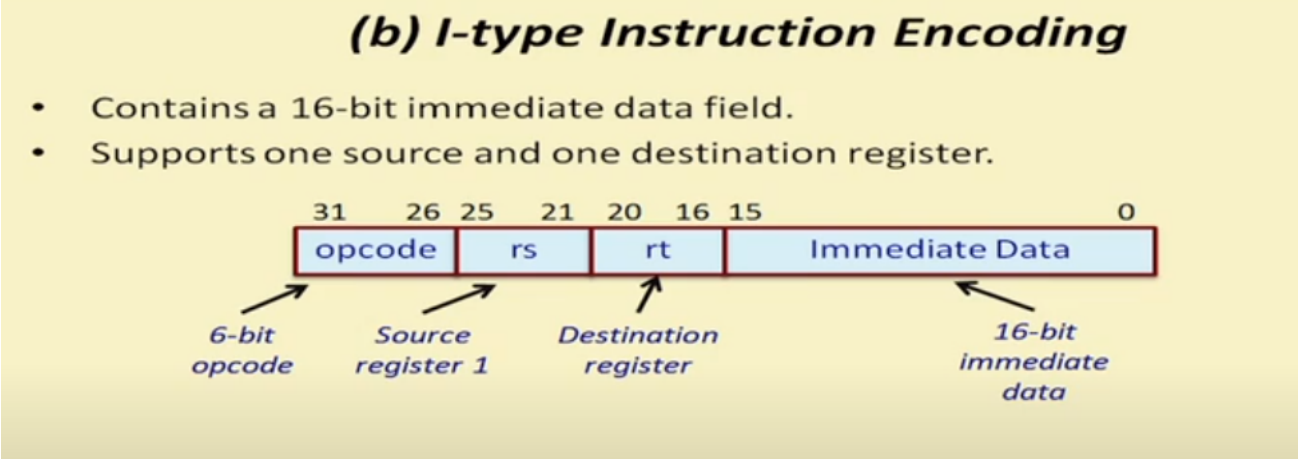


R-type instruction with opcode:

Instruction	Opcode
ADD	000000
SUB	000001
AND	000010
OR	000011
SLT	000100
MUL	000101
HLT	111111

Example: **SUB R6, R9, R5**

000001 00110 01001 00101 00000 000000

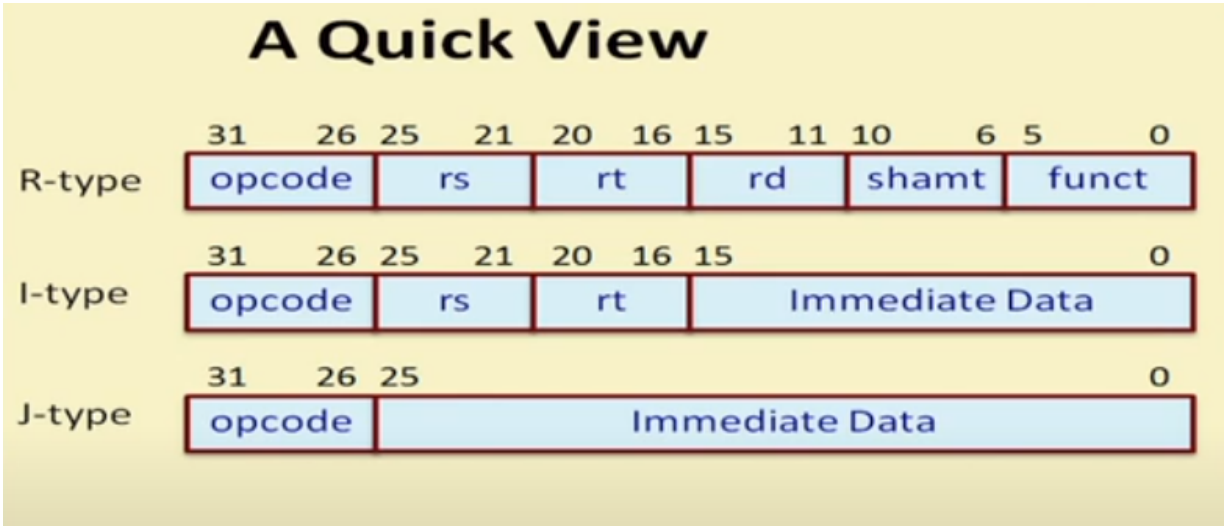


Immediate Instruction with opcode:

Instruction	Opcode
LW	001000
SW	001001
ADDI	001010
SUBI	001011
SLTI	001100
BNEQZ	001101
BEQZ	001110

Example: LW R20, 84(R9)

001000	01001	10100	0000000001010100
LW	R9	R20	Offset



Resource used: NPTEL