|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Intel 8080 | MIPS | ARM | Nvidia Fermi |
| RISC vs CISC | CISC | RISC | RISC | Looks RISC, feels CISC |
| Instruction Size | 8-bit | 32-bit | 32-bit+ | 64-bit |
| General Purpose Registers | 7 8-bit registers | 32 | 16 32-bit registers | 63 |
| Endian | Little | Big | Big | Little |
| Instruction Encoding | Variable | Fixed | Fixed | Fixed |
| Max Operands | 2 | 3 | 3 | 2 |
| Instruction Examples: |  |  |  |  |
| Add | LDA a (load a to memory)  ADD S (add register to A) | Add $d, $s, $t  ($d = $s + $t) | ADD R0, R1, R2  (R0 = R1 + R2) | IADD R1, R2 |
| Move | MOV D, S (move S into D) | Move $t2, $t3  ($t2 = $t3) | Mov R4, R5  (R5 is copied into R4) | Mov R1, R2 |

**Machine Code**

**Intel 8080**

**MIPS**

The MIPS data path unit consists of Instruction Memory, three ALUs, Register File, Control Unit (3 in, 8 out), a Sign Extend, Data Memory, and a Zero Extend. Instruction memory takes in the PC and outputs an instruction. The instruction is then fed to the control unit, register file, sign extend, and zero extend. The control unit is in control of memory to registry, write, branch, ALU Control, ALU source, Register destination, branch choice, and register write. At the end is the data memory which takes in write data, an ALU result that consists of the ALU control signal.

add $s0, $s1, $s2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| op | rs | rt | rd | shamt | funct |
| 000000 | 10001 | 10010 | 10000 | 00000 | 100000 |

**ARM**

Data Path Unit:

The main components of an ARM processor consist of the Register, Multiplier, Barrel Shifter, Arithmetic Logic Unit, and a Control Unit. In the design that was analyzed, the ARM processor consisted of 16 32-bit registers. Unlike other processors that have been looked at, ARM has each register of the same size.

**Nvidia Fermi**

IADD R1, R2

1100 000000 1110 000000 000000 0000000000000000000000 0000000000 010010  
        mod        reg0   reg1                  immea