

2 Stage MIPS32 Processor Implementation

Loay Rashid

AIM:

Implement a 2 stage MIPS processor, with different modules for fetch, ALU, decode, execute units. The processor must be able to execute a given set of MIPS32 instructions.

ARCHITECTURAL DESIGN:

ALU UNIT: Performs basic arithmetic and logic ops like add, xor, greater than, etc.

DECODE UNIT: Decodes the current instruction to get control signals (ALU_Ctr, mem_to_reg, etc)

FETCH UNIT: Update PC register depending on current instruction (PC increment by 4 or PC update for branch/jump).

EXECUTE UNIT: Executes instruction, updates all registers once ALU is done.

MAIN MEMORY UNIT:

Memory realized using 2-D arrays in verilog. Architecture used is Harvard Architecture (good for embedded systems).

DATA MEMORY:

- Little Endian
- Word Based (32 bits)
- 32x32 bits.

INSTRUCTION MEMORY:

- Byte based (8 bits)
- 128x8 bits.

REGISTERS:

- MIPS32 has 32 32bit registers on the processor
- 32x32 bits.

SUPPORTED INSTRUCTIONS:

- | | | |
|--------|---------|--------|
| • ADD | • SLLV | • ORI |
| • AND | • SRA | • XORI |
| • NOR | • SRAV | • BGTZ |
| • OR | • SRL | • BLEZ |
| • SUB | • SRLV | • BNE |
| • XOR | • ADDI | • LW |
| • SLT | • ANDI | • SW |
| • SLTU | • SLTI | • BEQ |
| • SLL | • SLTIU | • J |

- JAL

UNSUPPORTED MIPS32 INSTRUCTIONS:

- | | | |
|---------|--------|--------|
| • TLBP | • REM | • MLTO |
| • TLT | • MOD | • MTHI |
| • WAIT | • NAND | • MLFO |
| • PREFE | • BGT | • JALR |
| • PREF | • LB | • BAL |
| • PAUSE | • LH | • EXT |
| • MOVF | • SB | • LBE |
| • DIV | • SH | • LBU |
| • MULT | • TRAP | |

PROS:

- 2 stage pipelined architecture for higher throughput.
- Not too many changes required to add more MIPS32 instructions.
- Processor has a branch delay slot, can add independent instructions instead of NOOP.
- Can efficiently execute conditional/iterative statements.
- Processor Clock Frequency: 100 Mhz

CONS:

- Clock frequency can be reduced further by more efficient implementation+more pipelining
- Code is not modular, all modules (ALU, Control Unit, Fetch, etc) are in the same verilog file.
- Lower throughput due to branch delay slot.
- Cannot cleanly load instructions into the memory, need to change verilog code for that.
- Large number of MIPS32 instructions not implemented.