Lab 6: Single cycle datapath design

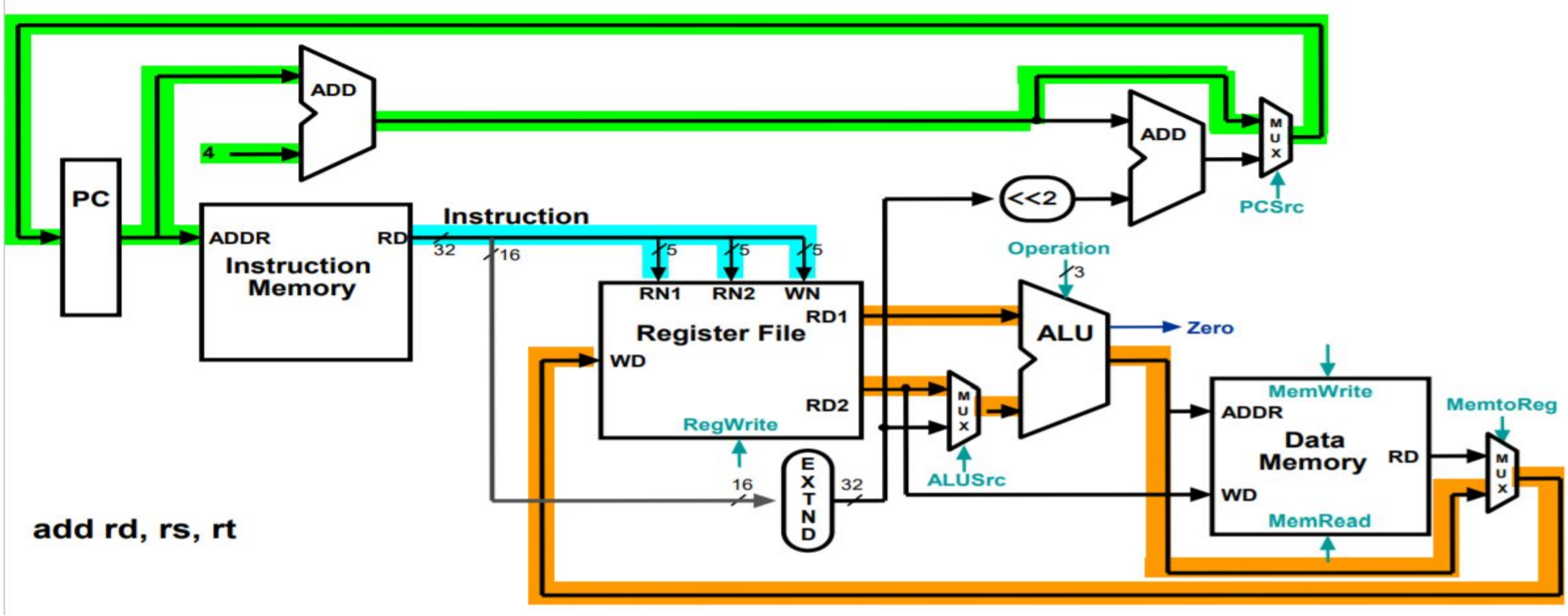
Learning outcomes:

- Modeling MIPS Single cycle datapath in Verilog
- Implementation of modules and integrating them

Single Cycle DataPath

6 bits	5 bits	5 bits	5 bits	5 bits	6 bits
op	rs	rt	rd	shamt	funct

R-Format



add rd, rs, rt

Modules to be integrated

Instruction Memory	Lab-6
Register File	✓
ALU	✓
Control Unit and ALU control	✓
Data Memory	Lab-6
Sign-Extender	Lab-6
Shifter	Lab-6
Adders (For Next PC address)	✓
Multiplexers	✓

✓: Done so far

Modeling Memory In Verilog

Instruction Memory

- Initialize Instruction memory with some R-format instructions.
- For now instruction memory size is 32 locations.

Data Memory

- Initialize data memory to zeroes [for load/store].

Sample data memory code

```
module instrucMem(pc,rst,instr);
        input[4:0] pc;
        input rst;
        output[31:0] instr;

        reg[31:0] mem[31:0];

        always@(pc)
                instr=mem[pc];

        always@(rst) begin
                if(rst) begin
                        mem[0]=32'h00000200;
                        mem[1]=32'h00000201;
                        mem[2]=32'h00000204;
                        mem[3]=32'h00000108;
                end
        end
endmodule
```

Important in initialization of memory

```
$readmemh("File", ArrayName, StartAddr,
EndAddr);

$writememh("File", ArrayName, StartAddr,
EndAddr)
```

Control Unit Design

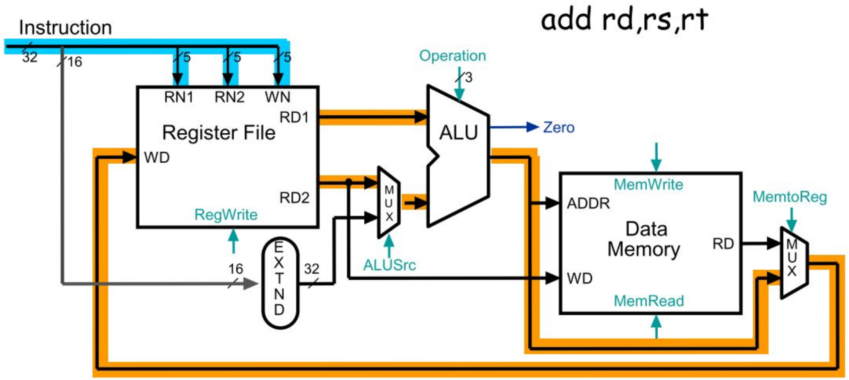
R-type:

Op-code	rs	rt	rd	shamt	funct
31:26	25:21	20:16	15:11	10:6	5:0

There are in total nine control signals [7 single bit and 2-bit ALU control].

- RegWrite
- ALUSrc
- ALUOp
- RegDst
- Memwrite-❌
- Memread- ❌
- Memtoreg- ❌
- PCSrc-❌

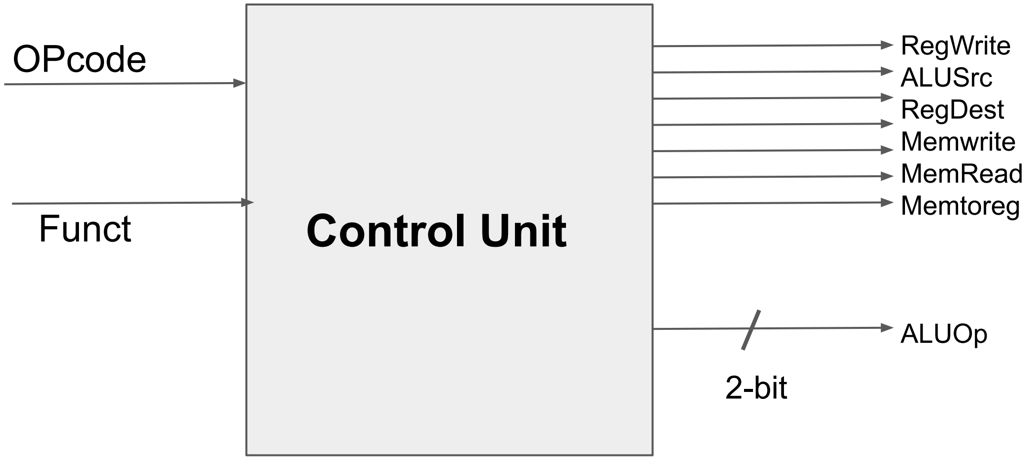
Datapath:
R-type Instruction



Control Unit Design

The two common implementation methodologies for a control unit are

- Combinational based
- PLA based



Sample code for Control Module
Sample testbench for Control Module.

Refer to the instruction set to get individual opcodes.

Sign Extender and Shifter

Sign Extender

- Used in Load & store type, and in Immediate type instructions whenever there is an involvement of immediate value.
- Sign-extend the MSB which is later added with PC to derive the new address.

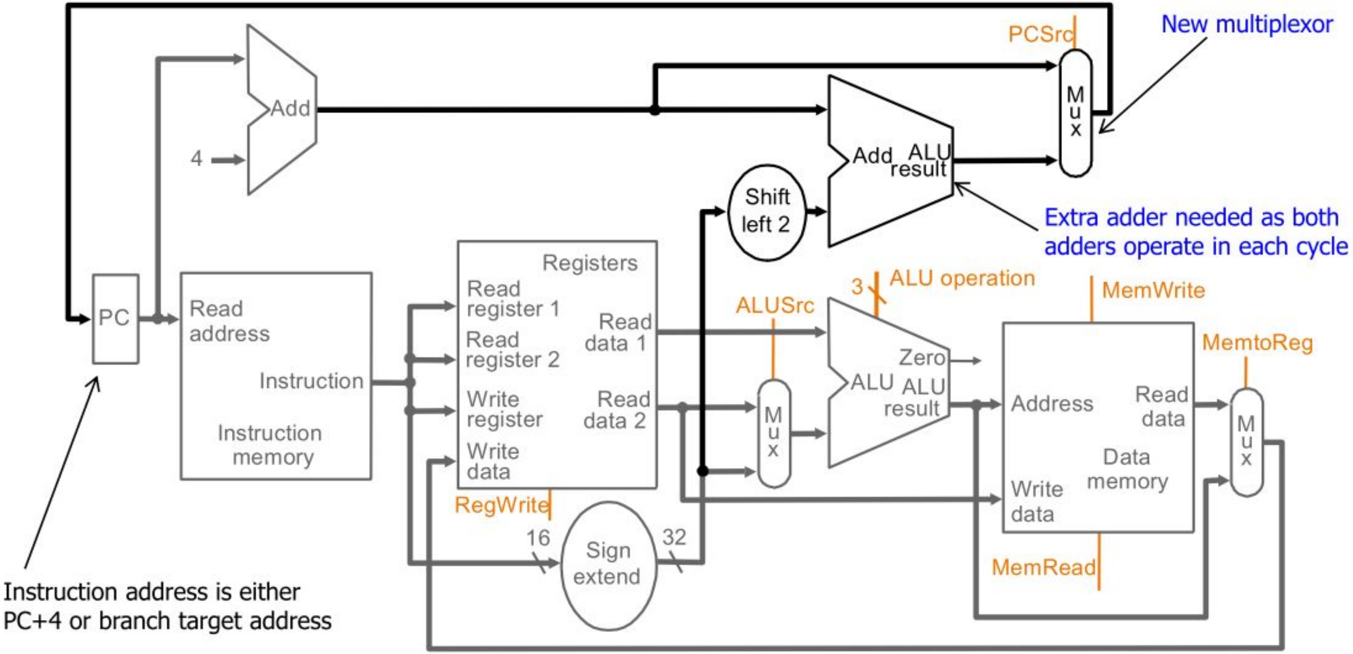
Ex:
```
extend_new = { {16{extend[15]}}, extend[15:0] };
```

Shifter

- Specifically used for Jump instruction.
- Memory is byte addressable

Ex:
```
outp = {inp[29:0],2'b00}
```

Putting it all together



Adding branch capability and another multiplexor