CSE 370 Homework 8 Solutions

1. x370 Model 2

Architectural Changes

The correct version of the x370 processor can be found in Figure 1. This now includes support for branch and load immediate instructions. To execute a branch, the control unit must tell the PC to load the new address from the instruction, and disable writing to the register file. To execute a load immediate instruction, the control unit must enable writing to the register file, and pass the immediate data from the instruction to the write port of the register file.

Control Unit

```
module control (inst ,rdAddrA ,rdDataB ,rdAddrB ,aluResult ,
                        wrAddr ,wrData ,regWrite ,aluOp ,loadPC );

// Parameters
parameter LDI = 5'h17;
parameter BR = 5'h0;
parameter BRZ = 5'h1;
parameter BRN = 5'h2;

// IO
output [2:0] rdAddrA;
output [2:0] rdAddrB;
output [2:0] wrAddr;
output [15:0] wrData;
output regWrite;
output [2:0] aluOp;
output loadPC;
input [15:0] inst;
input [15:0] rdDataB;
input [15:0] aluResult;

// branch instructions should not write to the register file
assign regWrite = inst[15];

// These bits of the instruction should always be hard-coded to
//  alu opcode, write address, and both read addresses, but we
//  don't always write to the register file
```

```
assign aluOp = inst[13:11];
assign wrAddr = inst[10:8];
assign rdAddrA = inst[5:3];
assign rdAddrB = inst[2:0];

// For a load immediate instruction, the data to be written to the
//  register file comes from the instruction.
assign wrData = (inst[15:11]==LDI) ? {{8{inst[7]},inst[7:0]} : aluResult;

// A new value gets loaded into the PC on a branch instruction.
assign loadPC = (inst[15:11]==BR) ? 1'b1 :
                                    ((inst[15:11]==BRZ)&&(rdDataB==16'h0)) ? 1'b1 :
                                    ((inst[15:11]==BRN)&&(rdDataB[15])) ? 1'b1 : 1'b0;
endmodule
```

Program Counter (PC)

```
module PC ( CLK ,Q ,D ,load ,reset );

// IO
output reg [5:0] Q;
input CLK;
input [5:0] D;
input load;
input reset;

// Program counter is synchronously reset
always @ (posedge CLK) begin
        if (reset)
                Q <= 6'b0;
        else if (load)
                Q <= D;
        else
                Q <= Q + 1'b1;
end

endmodule
```

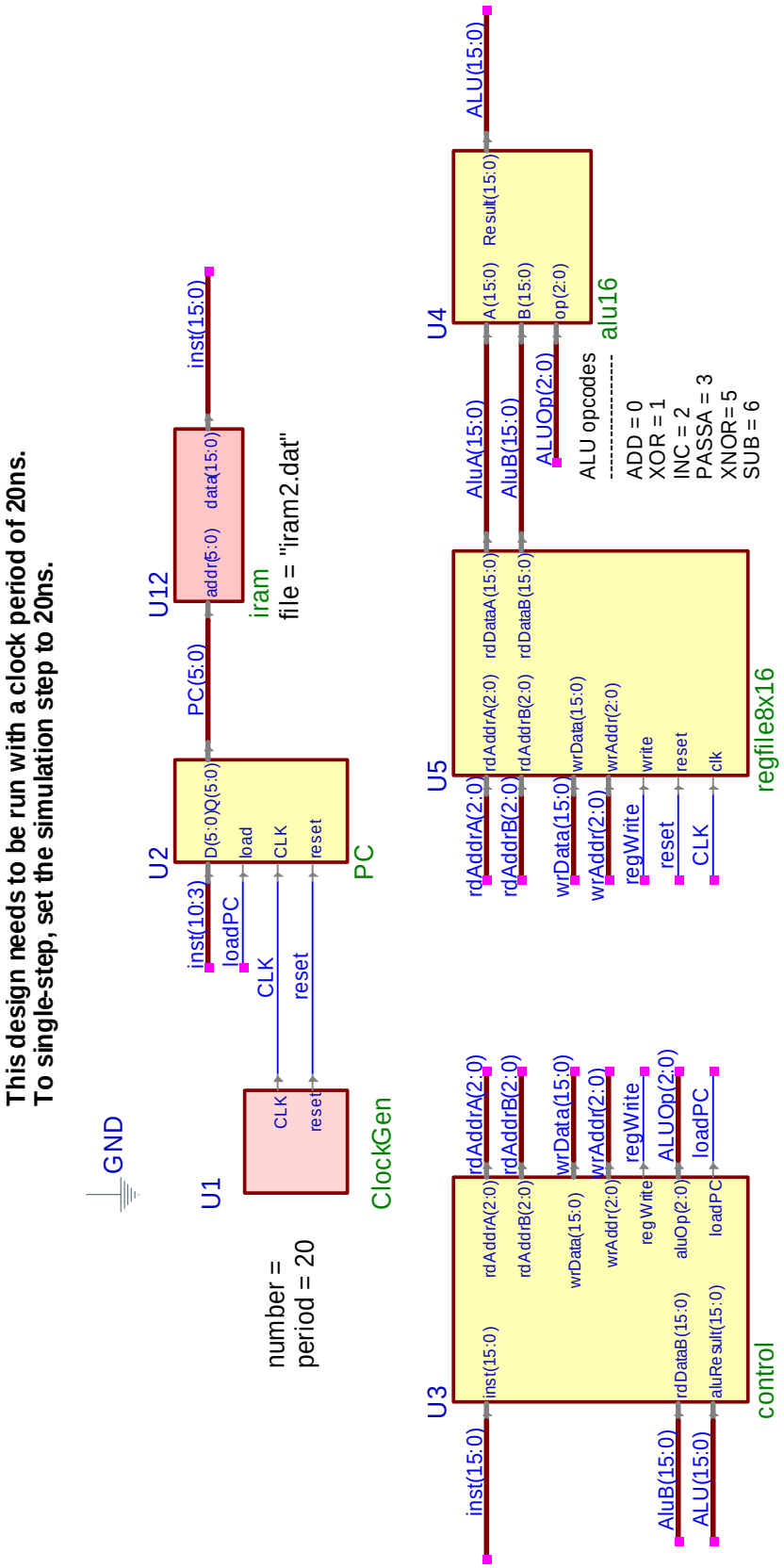

Figure 1:x370 Processor including support for load immediate and branching.

Multiplier Program for x370

```
// this program evaluates N * M and places result in R0
// N and M are arbitrary numbers, this solution uses N=4, M=6
10111_010___00000100    // R2 = 4 = N
10111_001___00000100    // R1 = 6 = M
10111_011___00000000    // R3 = 0 = i
10110_100_00_010_011    // R4 = R2 – R3 (ADDRESS 2)
00001_00001001___100    // BRZ 9, R4
00010_00001001___100    // BRN 9, R4
10000_000_00_000_001    // R0 = R0 + R1
10010_011_00_011_000    // R3 = INC R3
00000_00000010___000    // BR 2
00000_000___00000111    // NO-OP (ADDRESS 9)
```