# CS230: DLDCA

Project: IITB-RISC-22

Group:

Annam Saivardhan - 200050008<br>
Cheerla Vinay Kumar - 200050027<br>
Dendukuri Sandeep Varma - 200050032<br>
Dwarapudi Harshavardhan - 200050038

Designing a multi-cycle processor, IITB-RISC-22, whose instruction architecture is provided, with 8 registers. It is implemented in VHDL.

Let’s look at the components used in the design:

ALU (Arithmetic Logic Unit):

It takes two 16-bit vectors and performs various arithmetic and bitwise operations based on another input called the selector. It performs operations like addition, bitwise nand, etc. It outputs the vector after the operation and the cout and eq flags.

Immediate_6bit and 9bit:

These components do the sign extension of the 6bit and 9bit input vectors to 16bits.

Memory Unit:

This component outputs the data in the given address if the read input is 1 and writes data if the write input is 1 when rising_edge(CLK)

Register File:

This consists of 8 registers which we use according to the state we are currently in.

RISC:

This is the most important part of the machine as it integrates various sub-components we already implemented. It works as a Mealy machine.

This takes opcode, and values of c, and z registers as input to give the output.

There are 3 machine code instruction formats R, I, and J, and a total of 15 instructions.

States:

| | |
|---|---|
| S1<br>PC → Mem_A<br>Mem_D → IR<br>PC → ALU_A<br>+2 → ALU_B<br>ALU_C → T4 | S2<br>IR{8-0} → LE_16<br>LE_16 → T1 |

| | |
|---|---|
| S3<br>T1 → RF_D3<br>IR{11-9} → RF_A3<br>T4 → PC | S4<br>IR{11-9} → RF_A1<br>IR{8-6} → RF_A2<br>RF_D1 → T1<br>RF_D2 → T2 |
| S5<br>T1 → ALU_A<br>T2 → ALU_B<br>ALU_C → T1<br>ALU_Z → Z<br>ALU_CA → C<br>Subtraction | S6<br>T1 → RF_D3<br>IR{5-3} → RF_A3<br>T4 → PC |
| S7<br>IR{11-9} → RF_A1<br>RF_D1 → T1<br>IR{5-0} → SE_16<br>SE_16 → T2 | S8<br>IR{8-6} → RE_A2<br>RF_D2 → T2<br>IR{5-0} → SE_16<br>SE_16 → T1 |

| | |
|---|---|
| S9<br>IR{4-9} → RE_A1<br>RE_D1 → T1<br>‘000’ → T2 | S10<br>T1 → Mem_A<br>Mem_D → T3 |
| S11<br>T3 → RF_D3<br>T2{2-0} → RF_A3<br>T2 → ALU_A<br>T1 → ALU_B<br>ALU_C → T2 | S12<br>T1 → ALU_A<br>T2 → ALU_B<br>ALU_C → T4<br>T4 → PC |
| S13<br>T2 → ALU_A<br>+1 → ALU_B<br>ALU_C → T2 | S14<br>IR{8-6} → RE_A1<br>RE_DA → PC |
| S15<br>T2 → ALU_A<br>+1 → ALU_B<br>ALU_C → T2 | S16<br>T2{2-0} → RE_A2<br>RF_D2 → T3<br>T1 → Mem_A<br>T2 → Mem_D |
| S18<br>PC → ALU_A<br>IR{5-0} → SE_16<br>SE_16 → ALU_B<br>ALU_C → PC | S19<br>T1 → Mem_A<br>Mem_D → RF_DC3<br>IR{11-9} → RF_A3<br>T4 → PC |
| S20<br>IR{11-9} → RF_A1<br>RF_D1 → T2<br>T1 → Mem_A<br>T2 → Mem_D | S21<br>T1 → RF_D3<br>IR{8-6} → RF_A3<br>T4 → PC |

| S22 | S23 |
|---|---|
| PC → RF_D3<br>IR{11-9} → RF_A3 | IR{11-9} → RF_A2<br>RF_D2 → T1<br>T1 → ALU_A<br>Imm_9 → ALU_B<br>ALU_C → PC |

Flow Diagram for various operations:

ADD, ADC, ADZ, ADL:

S1 → S4 → S5 → S6

ADI:

S1 → S7 → S5 → S21

NDU, NDZ, NDC:

S1 → S4 → S5 → S6

LHI:

S1 → S2 → S3

LW:

S1 → S8 → S5 → S19

SW:

S1 → S8 → S5 → S20

LM:

S1 → S9 → S10 → S11 → S12 (if T2 < 8) → S10

SM:

S1 → S9 → S16 → S13 → S12 (if T2 < 8) → S16

BEQ:

S1 → S4 → S5 → S18

JAL:

S1 → S22 → S14

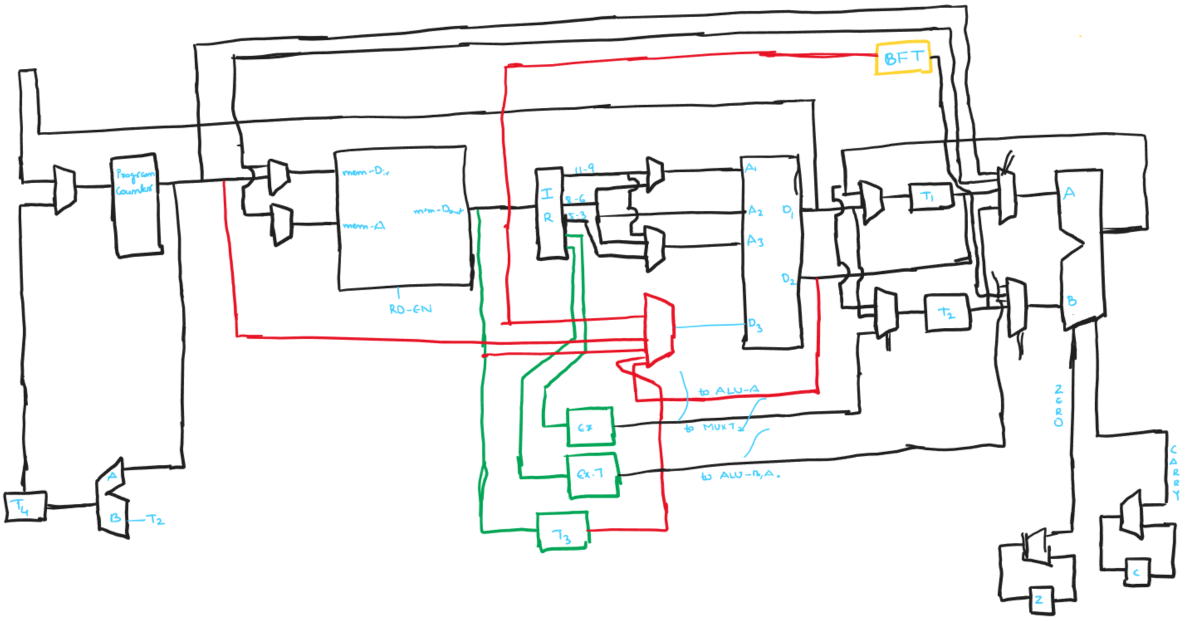
JLR:

S1 → S13 → S14

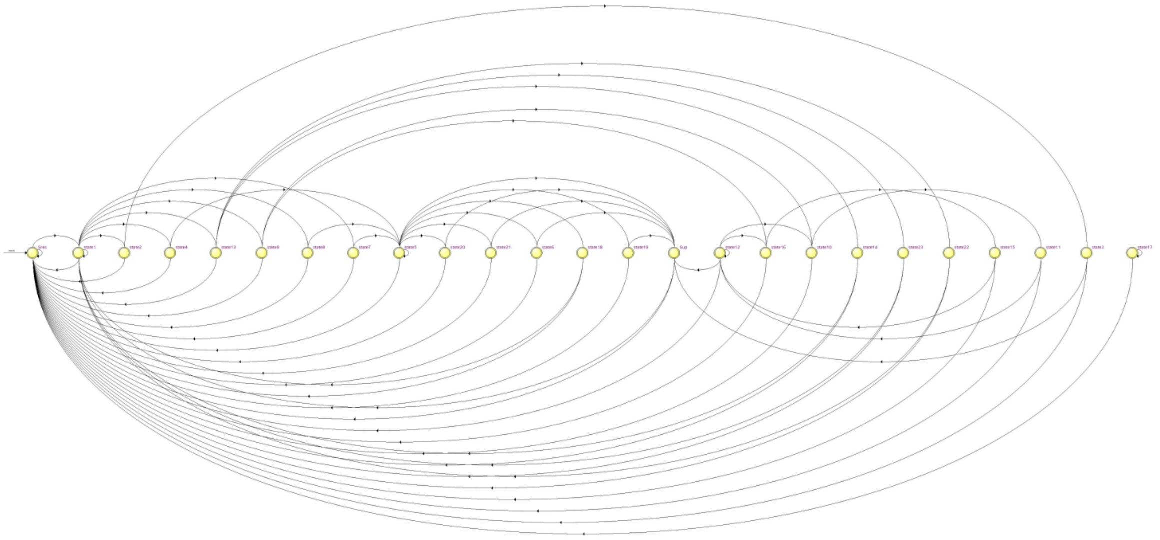
JRI:

S1 → S13 → S23

General Data Flow:



# State Machine Diagram:


Sres
state1
state2
state4
state13
state9
state8
state7
state5
state20
state21
state6
state18
state19
Sup
state12
state16
state10
state14
state23
state22
state15
state11
state3
state17