EE309

IITB RISC - 22

Multi-cycle processor

Prof. Virendra Singh

Anirudh Kansal	Keshav Singhal	Mohit Kedia	Nishant Thakre
20d070013	20d070047	200020079	200070051

Components:

ALU:

1. ir_out (5 downto 0) is made to be of size 16 bits by padding with zeros. (s6_out)
2. ir_out (8 downto 0) is made to be of size 16 bits by padding with zeros. (s9_out)
3. alu_a_mux uses s6_out to make alu_a_in
4. alu_b_mux uses s9_out and s6_out to make alu_b_in
5. Now, in ALU block takes alu_b_in and 00...001 and multiplexes using bit_en to give output at alu_b_final_in
6. ALU gets an input bit op_code. If this bit is zero, then addition takes place between alu_a_in and alu_b_final_in and the bit c_out takes the carry. If this bit is high, bitwise NAND takes place and c_out bit is low.
7. alu_out takes the result of this operation.
8. If the bits (c(0) to c(15) are all zero, then the z_out is high (c(16) does not play a role here)) (z_out = zero flag).

Memory:

Allows read and write operations to memory

xor_operation :

Computes xor of two 16 bit numbers

PadZero

Takes 8-bit number as input ir_8_0 and gives output as a 16-bit number as ir_8_000000000 I.e. ir_8_0 followed by 8 zeroes

nine_bit_pad
Takes 8-bit number ir_8_0 as input gives output as 16-bit number 0000000ir_8_0

six_bit_pad
Takes 6-bit number ir_5_0 as input gives output as 16-bit number 0000000000ir_5_0

register_b
Used in c and z flags, output 0 when reset is set or if enable is set the input bit is written on the flag

register_a
Similar to register_b but instead of taking 1-bit output and input, it works on 16 bits I/O

bit_shift
Shifts all the bits to the left by one position in a cyclic manner

Register_file
Based on reset and wr_en it allows to reset the memory location or write to it and also allows memory access using rf_d1 and rf_d2

Priority_Encoder
Takes a 1-bit input in the form of t4 and outputs a 3-bit vector in the form of decode_in

mux_2to1
Takes input x0,x1 each of which is a 16-bit number; takes 1 selection bit and gives one of x0,x1 as output

mux_4to1
Takes input x0,x1,x2,x3 each of which is a 16-bit number; takes 2 selection bits and then gives one of x0,x1,x2,x3 as output

mux_4to1_3bit

Takes input x0,x1,x2,x3 each of which is a 3-bit number; takes 2 selection bits and gives one of x0,x1,x2,x3 as output

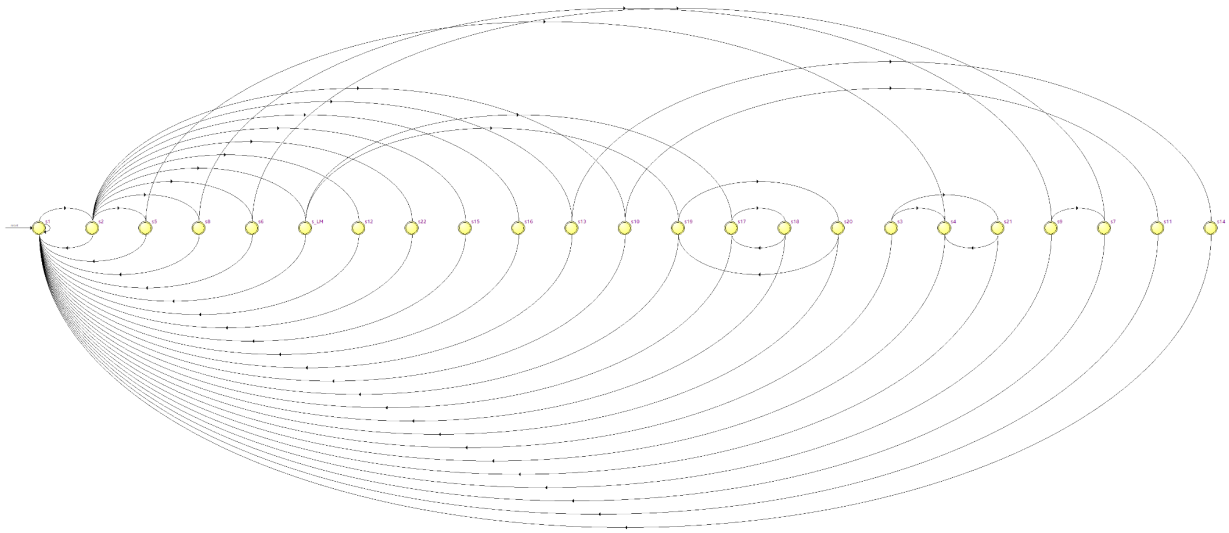
mux_3to1_3bit

Takes input x0,x1 each of which is a 3-bit number; takes 1 selection bit and gives one of x0,x1 as output

Decoder

Takes 3 bit decode_in and returns “16 bit” decode_out

State Machine View:



Total No. of states - 23

RTL View: