Documentation for Lab 5 part 5 New Additions

bne – Branch Not Equal

This instruction moves the PC by a given amount if the given registers do not hold equal values.

Instruction Structure : bne 0x(immediate_value) (register1) (register2)
example : bne 0x02 1 2

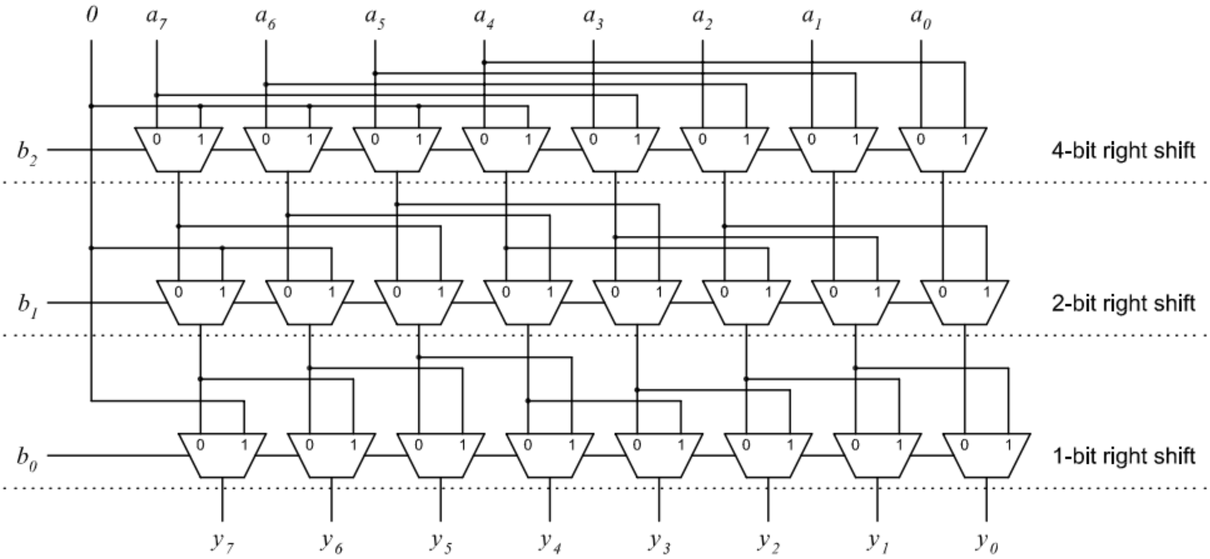
opcode : 0000_1000

changes : bne was implemented just like beq, a new CU signal was introduced and it was high for bne, according to the opcode. Inverse of the ALUZERO signal was taken and from that and the new CU signal, PC was changed using the value extracted from Immediate value.

srl – Logical Shift Right

This instruction right shifts a value in a specified register by the amount specified in the immediate value, and stores in a specified register.

Instruction Structure	: srl (store reg) (value taken reg) 0x(immediate_value)
example	: srl 4 1 0x02 take the value in reg1 right shift it by 2 bits and store it in reg4
opcode	: 0000_1001
changes	: a new module was added to the ALU and it has 24 muxes. A new ALU opcode was introduced and it is selected in the CU when necessary. Mux diagram is shown below.
Timing	: to simulate the delays, 2 time units of delay was added to the new module inside ALU



8-bit logical right shifter.

sll – Logical Shift Left

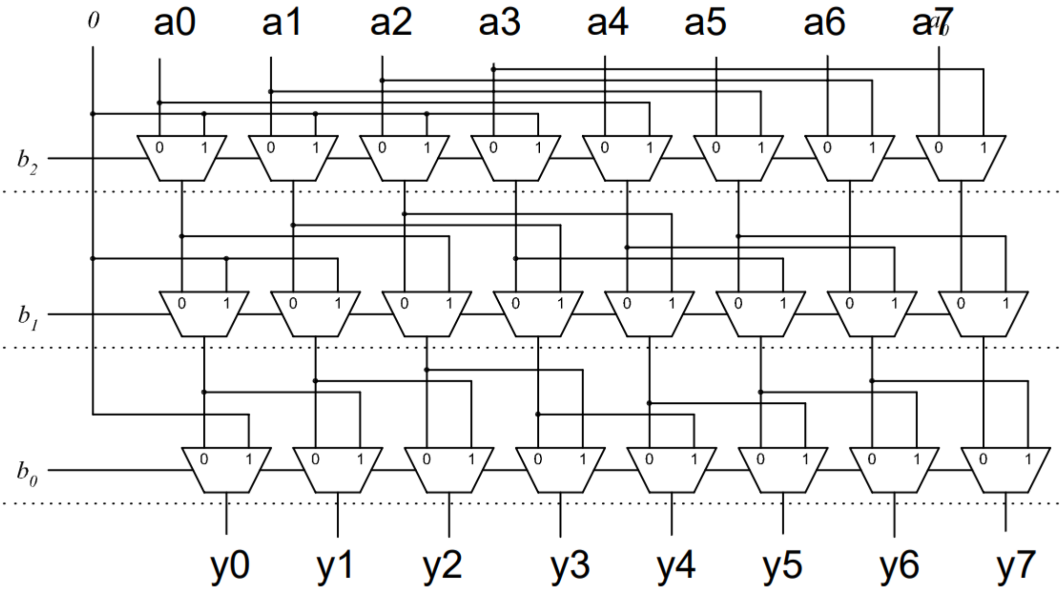
This instruction left shifts a value in a specified register by the amount specified in the immediate value, and stores in a specified register.

Instruction Structure : sll (store reg) (value taken reg) 0x(immediate_value)
example : sll 4 1 0x02
take the value in reg1 left shift it by 2 bits and store it in reg4

opcode : 0000_1010

changes : a new module was added to the ALU and it has 24 muxes. A new ALU opcode was introduced and it is selected in the CU when necessary.
Mux diagram is shown below.

Timing : to simulate the delays, 2 time units of delay was added to the new module inside ALU



8-bit Logical Left Shifter

sra – Arithmetic Shift Right

This instruction right arithmetic shifts a value in a specified register by the amount specified in the immediate value, and stores in a specified register.

Instruction Structure : sra(store reg) (value taken reg) 0x(immediate_value)

example : sra 4 1 0x02
take the value in reg1 arithmetic right shift it by 2 bits and store it in reg4

opcode : 0000_1011

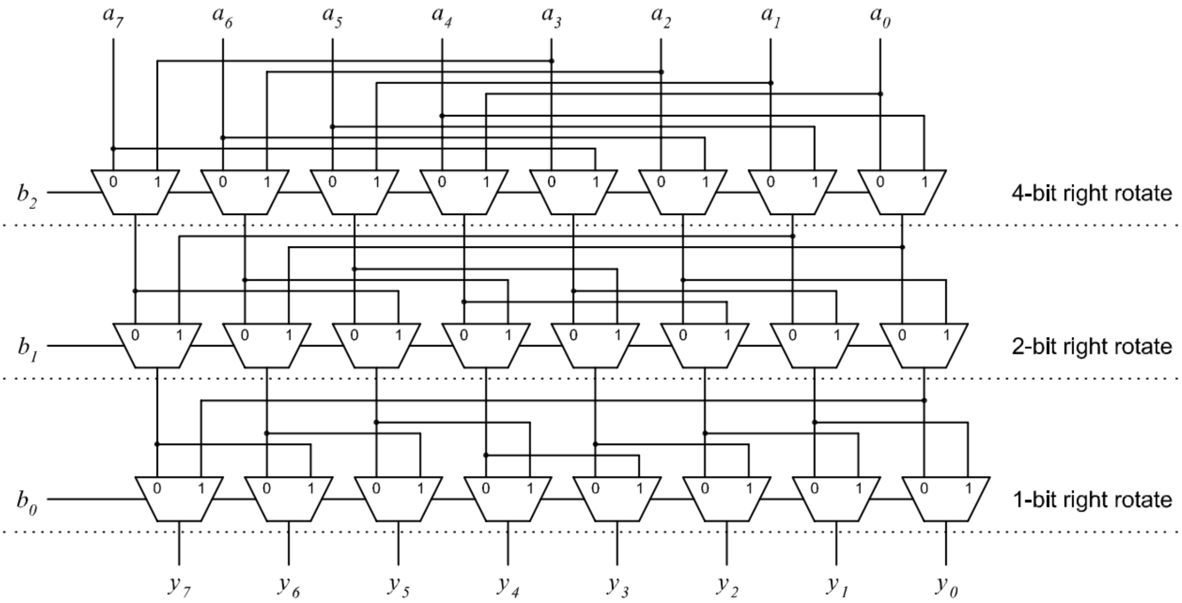
changes : a new module was added to the ALU and it has 24 muxes. A new ALU opcode was introduced and it is selected in the CU when necessary according to the Opcode.

Timing : to simulate the delays, 2 time units of delay was added to the new module inside ALU

ror – Rotate Right

This instruction right rotates a value in a specified register by the amount specified in the immediate value, and stores in a specified register.

Instruction Structure	: ror(store reg) (value taken reg) 0x(immediate_value)
example	: ror 4 1 0x02 take the value in reg1 rotate right it by 2 bits and store it in reg4
opcode	: 0000_1100
changes	: a new module was added to the ALU and it has 24 muxes. A new ALU opcode was introduced and it is selected in the CU when necessary. Mux diagram is shown below.
Timing	: to simulate the delays, 2 time units of delay was added to the new module inside ALU


a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0
b_2
b_1
b_0
0 1
4-bit right rotate
2-bit right rotate
1-bit right rotate
y_7 y_6 y_5 y_4 y_3 y_2 y_1 y_0

8-bit right rotator.