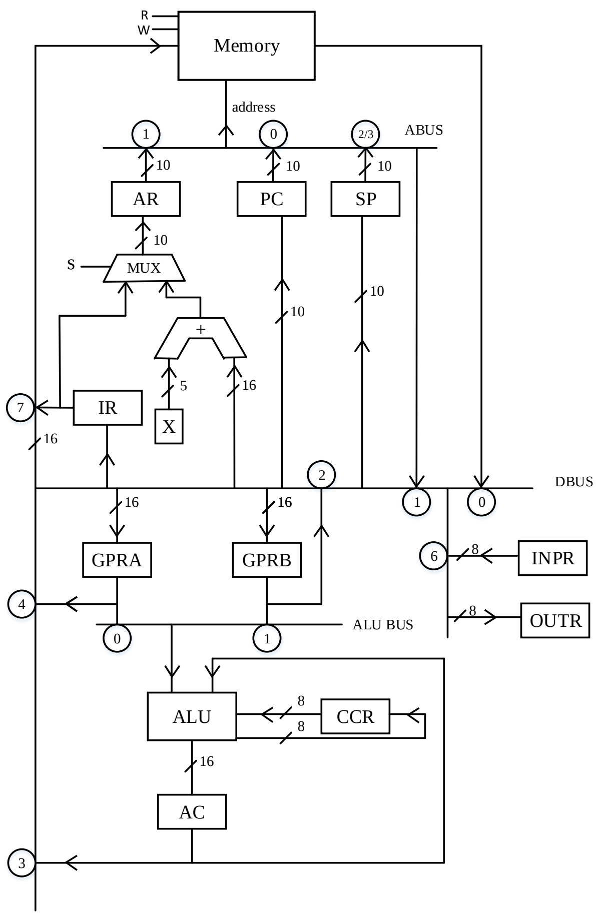
BLG222E - Computer Organization
Project 3

Add hardwired control unit to the following computer. This is the same architecture that you have designed in Project #2.



There are 11 registers in the computer: PC, AR, SP, X, GPRA, GPRB, AC, IR, CCR, INPR, and OUTR. Each register should have control signals of load (LD), clear (CLR) and increment (INC). The register lengths are listed below.

Register	length
GPRA, GPRB, AC, IR	16-bit
AR, PC, SP	10-bit
INPR, OUTR, CCR	8-bit
X	5-bit

The multiplexer in the input of the AR works as follows:

S	MUX Output
0	IR(9-0)
1	X+DBUS(9-0)

When the 16-bit value in DBUS is loaded by PC, SP, or OUTR only the lower bits are transferred.

The control signals for ABUS are A_1 and A_0. The registers write to the ABUS according to the following table:

Control signal		Register to write ABUS
A_1	A_0	
0	0	PC
0	1	AR
1	dont care	SP

The control signals for DBUS are D_2, D_1 and D_0. The registers write to the DBUS according to the following table:

Control signal			Register to write DBUS
D_2	D_1	D_0	
0	0	0	Memory
0	0	1	ABUS
0	1	0	GPRB
0	1	1	AC
1	0	0	GPRA
1	0	1	X
1	1	0	INPR
1	1	1	IR

The control signal for ALUBUS is M, and the registers write to the ALUBUS according to the following table:

Control signal M	Register to write ALUBUS
0	GPRA
1	GPRB

The memory has 2 control signals read (R) and write (W).

The instruction format for this computer is as follows:

15		0
2-bit	4-bit	10-bit
addr. mode	opcode	operand address

First two bits are the addressing mode bits. There are 4 different addressing modes:

Addr. mode		Effective address (EA)	mode
0	0	$EA \leftarrow IR(9-0)$	direct
0	1	$EA \leftarrow M[IR(9-0)]$	indirect
1	0	$EA \leftarrow IR(9-0) + X$	indexed
1	1	$EA \leftarrow SP$	stack

Inside the control unit, the opcode bits of the IR register is connected to an opcode decoder that generates signals from K_0 to K_{15}. In addition there are two flags (R_1 and R_0) that are connected to the address mode bits in IR: $R_1 \leftarrow IR(15)$ and $R_0 \leftarrow IR(14)$. Finally, there is a 4-bit sequence counter in the control unit that is connected to a decode that generates timing signals of T_0, T_1, ..., T_{15}.

The control unit should be able to fetch an instruction from the memory, decode the instruction. The control unit should be able to execute the following instructions. Note that this is only a subset of a complete instruction set that is assigned considering the time given for this project.

Instruction subset:

Symbol	Opcode (binary)	Description
LDA	0001	GPRA ←M[EA]
LDB	0010	GPRB ←M[EA]
STA	0011	M[EA] ←GPRA
STB	0100	M[EA] ←GPRB
ADDA	0101	AC ←AC + GPRA
ADDB	0110	AC ←AC + GPRB
BUN	0111	PC ←EA
BZE	1000	if Z=1 then PC ←EA
BNE	1001	if N=1 then PC ←EA
XCH	1010	Exchange values in GPRA and GPRB
CLRAC	1011	AC ←0
CLRX	1100	X ←0
INCX	1101	X ←X+1
INCB	1110	GPRB ←GPRB+1
LDSP	1111	SP←M[EA]

Note: Ignore the interrupt cycle.

Consider the following assembly code. In this code, `x` means do not care.

```
 ORG  0x100
      xx CLRAC x
      xx CLRX  x
      00 LDB   0
LOOP: xx INCX  x
      10 LDA   0
      xx ADDA  x
      xx INCB  x
      00 BZE   LOOP
      xx INCX  x
      10 STA   0
```

Find the machine code for this program and write it as address & instruction pairs. Describe the purpose of this code. Your computer should be able to execute this code.

If the memory is as follows before the execution of this code, find the state of the memory after the code execution.

Memory before code execution:

```
0 - 1111 1111 1111 1011
1 - 0000 0000 0000 0001
2 - 0000 0000 0000 0011
3 - 0000 0000 0000 0101
4 - 0000 0000 0000 0111
5 - 0000 0000 0000 1001
6 - 0000 1000 0010 0000
7 - 1010 1111 0110 1100
8 - 0000 1111 0011 0000
```

For midterm 2:

It is **strongly** recommended to review this project and architecture before Midterm 2.

Groupwork:

Group work is expected for this project. Same group (from the previous project) of students should design together. You might be asked to make a 15-minute demonstration of your design with a few test cases.

What to turn in:

Implement your design for the register and simple computer in **logisim** software, upload a single compressed (zip or rar) file to ninova before the deadline. Only one student from each group should submit the project file. This compressed file should contain:

- the student number&names of the students in the group
- design (.circ) file for the simple computer with hardwired control unit.
- machine code of the given code in address & instruction pairs.
- description of the purpose of this code.
- memory state after the code execution in address & instruction pairs. Only data in addresses between 0–8 are required.
- a short report that lists of control inputs and corresponding functions of the simple computer