CE869: "FPGA-based Microprocessor Design"

(Assignment 2)

Submitted as part of the requirements for:

UNIVERSITY OF ESSEX

CE869 HIGH LEVEL DIGITAL DESIGN

MSc Electronic Engineering

Department of CSEE

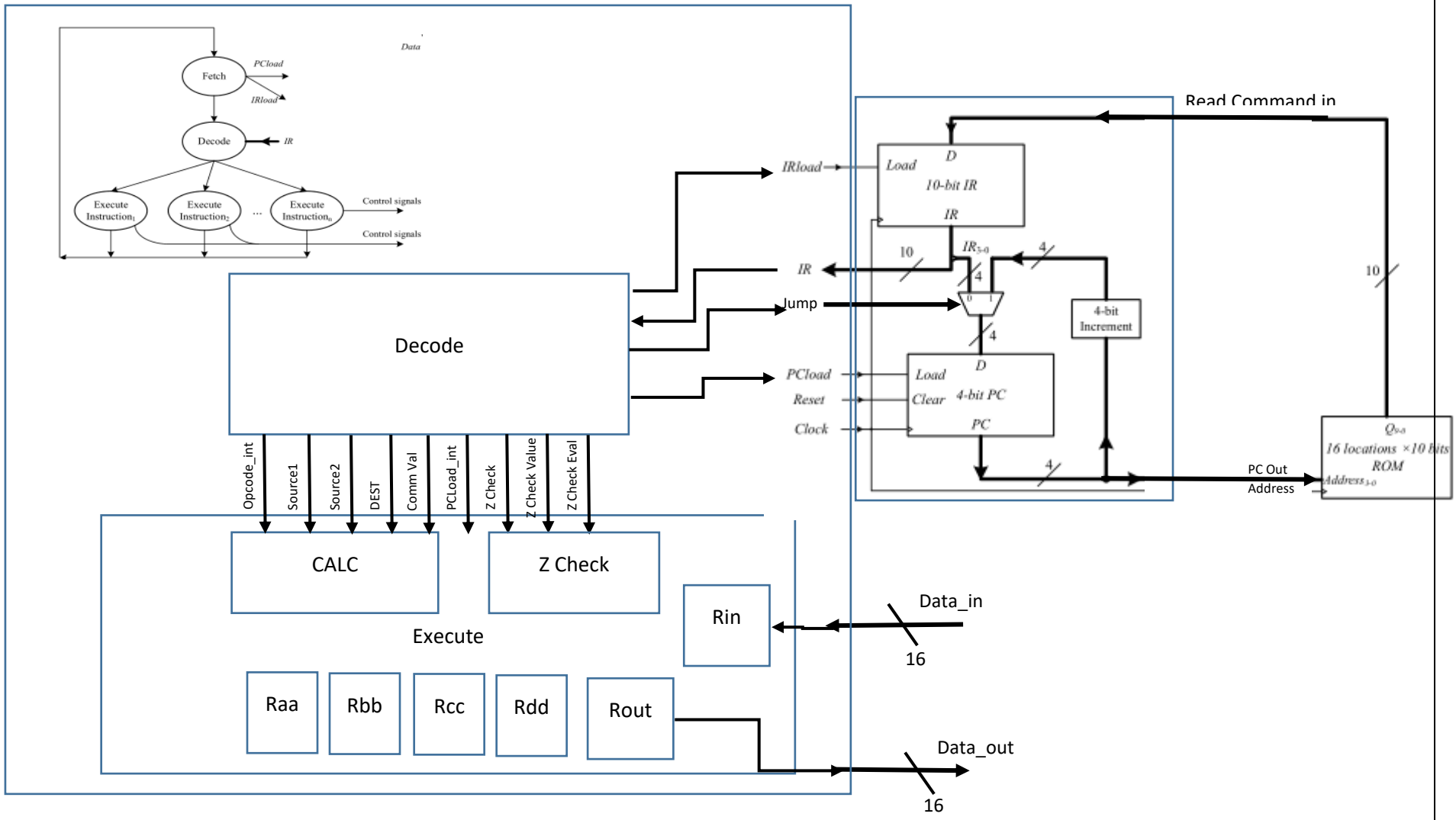
Name: FALGUNI VASAVA

Regd. No: 2006617

Module Supervisor: Dr Xiaojun Zhai

Date: 02/04/2021

CPU Design


Fetch
PCload
IRload
Decode
IR
Execute Instruction1
Execute Instruction2
Execute Instructionn
Control signals
Control signals
Data
Read Command in
IRload
Load
D
10-bit IR
IR
10
IR3-0
4
IR
Jump
4-bit Increment
PCload
Reset
Clock
Load
Clear
4-bit PC
PC
Q3-0
16 locations ×10 bits ROM
Address3-0
PC Out
Address
Decode
Opcode_int
Source1
Source2
DEST
Comm Val
PCLoad_int
Z Check
Z Check Value
Z Check Eval
CALC
Z Check
Execute
Rin
Data_in
16
Raa
Rbb
Rcc
Rdd
Rout
Data_out
16

The System is divided into following parts –

1. Program ROM.
2. Fetch Unit.
3. Control & Decode Unit.

The main operation of each of parts are given below –

1. **Program ROM:** This carry the Program.
    a. It takes the PC or Address as input and give out 10-bit pre-programmed data from the respective memory location.

2. **Fetch Unit:** Fetch means reading the instruction from memory. The instruction is stored in in instruction memory. This carries the program fetching circuit. The actual processing happening in this unit is given below –
    a. It takes the instruction like IRload, PCload and Jump from the Control and decode circuit After taking this instruction is performing the following operations.
        i. IRLoad: when IR load is asserted then following action is happed -
            1. the data/ instruction is coming from ROM which is loaded into the internal memory its nothing it is flipflop.
            2. The stored instruction is giving out to Control & decode circuit for further processing.
        ii. PCload: once PCLoad is asserted, then following happens –
            1. The Incremented (A+1) or Jump(J) address is loaded into the PC register which is driven out on the PC Bus to the Memory.
        iii. Jump: Once Jump is asserted, the New Jump address fetched out from the incoming last command (which is already in IR register) is loaded to the PCLoad register & This new PC points to new loaded jumped address.

3. Control and decode circuit: Its carry following circuits –
    a. Control FSM.
    b. Instruction Decode.
    c. Command Process.
    d. Calculation Unit.
    e. Execution Unit.

4. The Control FSM:
    a. The Control FSM has following basic 4 states:
        i. IDLE
            1. When in reset the system stays in IDLE state.
            2. Once reset is asserted, the system moves to Fetch to take the first command from the memory
        ii. Fetch
            1. The state here asserts the IRLoad signal which do following things –
                a. In fetch, the fetch unit loads in incoming 10bit instruction from the ROM and stores into its local memory.
                b. The stored instruction is given out to the Control and decode circuit for processing.
        iii. Decode
            1. In decode state, the incoming 10-bit instruction is divided into following parts
                a. 4-bit Command => these commands are broken into following internal commands for further processing –

OPCODE	Command	Opcode Internal	Internal Opcode name
0000	HALT	000	
0001	MOV	000	Load to destination
0010	IN	000	Load to destination
0011	OUT	000	Load to destination
0100	NOT	001	Load not to destination
0101	JMP	010	Jump
0110	JNZ	010	Jump
0111	JN	010	Jump
1000	LT	000	Load to destination
1001	INC	011	Dest = Source1 + source2
1010	DEC	100	Dest = Source1 – source2
1011	ADD	011	Dest = Source1 + source2
1100	SUB	100	Dest = Source1 – source2
1101	AND	101	Dest = Source1 **AND** source2
1110	OR	110	Dest = Source1 **OR** source2
1111	MOV2	000	Load to destination

            2. these decoded commands generate following signals as well which are set for different decoding –
                a. opcode_int: Internal opcode as per table above.
                b. source1_sel, source2_sel, dest_sel:
                    i. There are four internal registers, One Input Port, One output port and one input coming from instruction IR.

ii. Each of these ports have given internal address which is used in Source1, source2 and destination address register.
iii. All these Source1, source2 and destination address register are of 4 bits.
iv. Following is the encoding used on these internal register addresses –

Source1, source2 and destination address	**Internal 16 bit register NAME**
0000	AA
0001	BB
0010	CC
0011	DD
0100	IR(3 downto 0)
1000	Input
1111	Output

c. comm_val: used to say that decoding command is valid to be executed.
d. PCload_int: it is a one-bit signal to indicate PC Load command has been identified during decoding.
e. Z_Check: it is a one-bit signal to indicate Z evaluation command has been identified during decoding.
f. Z_check_val: it is a one-bit signal to indicate value of Z to be used while evaluating the command has been identified during decoding.
g. Z_val_eval: it is a one-bit signal to indicate Z evaluation while command execution has been identified during decoding.

iv. Execute: the execution stage is divided into following two major stages where above decoded signals and internal opcodes are used. It is majorly controlling following stages –
1. Calculation Unit.
a. Here all the data coming from different sources as explained above are fed into respective selected processing path and the calculated 16-bit output is given to the respected selected output port as selected by destination port in the respective commands.
2. Command Execution Unit
a. The Execution unit carry following parts –
i. Data Storage
1. If the opcode_int is pointing to internal storage, then calculated data is stored to respective storage register.
ii. Next Command Fetch control.

1. If the opcode_int is instructing a JUMP or JUMP based on Z then respective signals to control the JUMP are invoked.

iii. Output control

1. If the opcode_int is pointing to output port, then calculated data is pointed to output ports.

Output picture: