KATHMANDU UNIVERSITY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

PROJECT REPORT

Design of a 16 bit Computer

Submitted by:

Manish Karn (22035)

Department of Electrical and Electronics Engineering

Submitted to:

Mr. Dhiraj Shrestha

Department of Computer Science and Engineering

August, 2015

Table of Contents

INTRODUCTION

Computer Architecture is concerned with the structure and behavior of the various functional modules of the computer and how they interact to provide the processing needs of the user. Computer organization is concerned with the hardware components are connect together to form a computer system. Computer design is concerned with the development of the hardware for the computer taking in the consideration a given set of specification. A computer system consists of peripheral devices, memory units and central processing unit (CPU). The central processing unit consists of arithmetic and logical unit, control unit, the primary memory and different registers to process the instructions given to it. These units are constructed from digital components such as registers, decoders, arithmetic elements, and logical gates. Digital modules is defined by the registers they contain and the operations that are performed on the data stored in them. The operations executed on data stored in registers are called micro-operations. A micro-operation is an elementary operation performed on the information stored in one or more registers. The result of the operation may replace the previous binary information of a register or may be transferred to another register. The internal organization of a digital system is defined by the sequence of micro-operations it performs on data stored in its registers. A computer follows the instruction set stored in the memory to perform specified tasks. The instruction code is divided into parts, each having its own interpretation. The basic computer design of a 16 bit memory element consists of 9 registers, 3 to 8 decoder and instruction sets.

BLOCK DIAGRAM OF THE SYSTEM FLOW OF 16 BIT COMPUTER SYSTEM

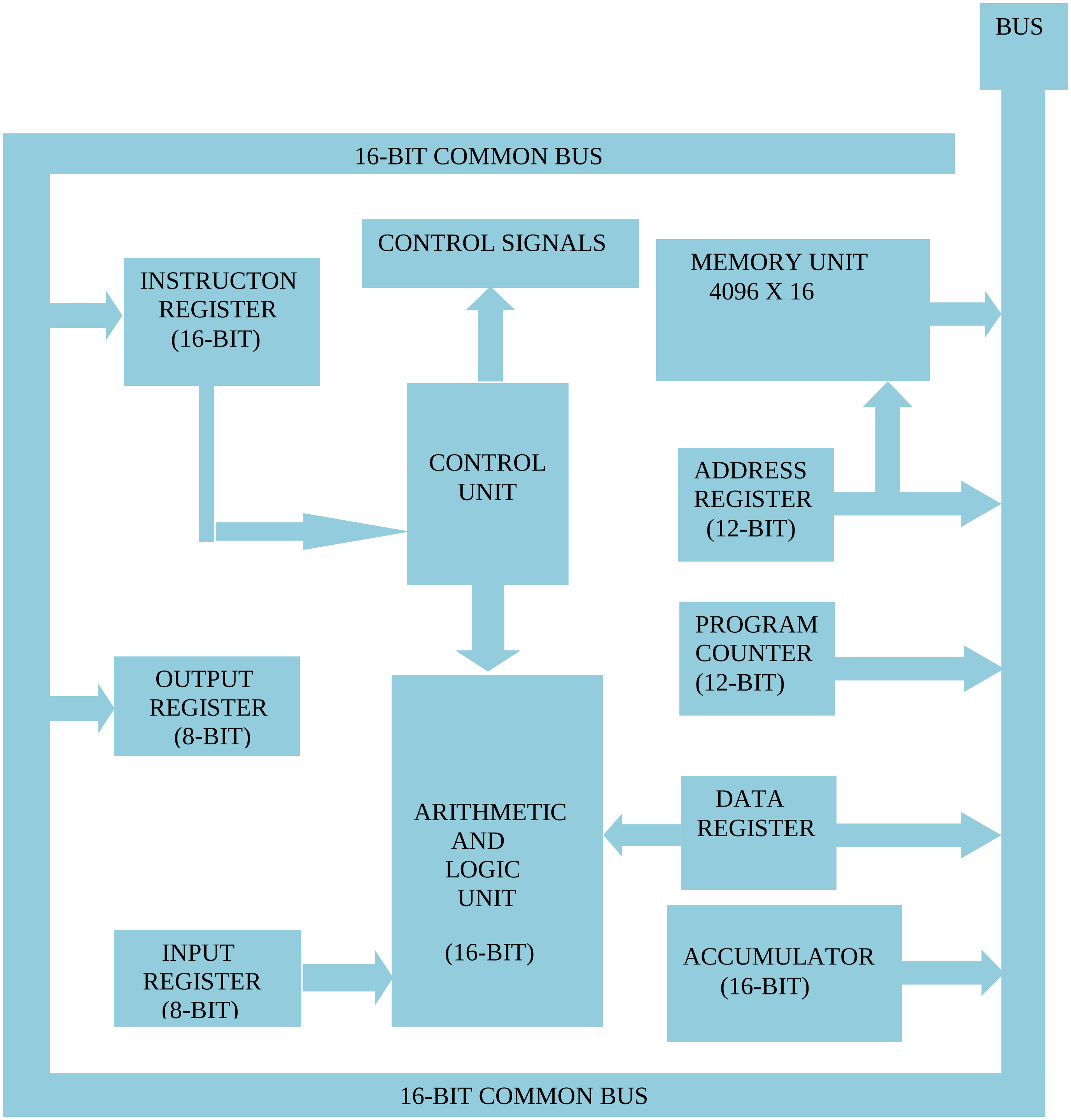


Fig: A 16 bit bus system

COMPONENTS USED IN THE 16-BIT SYSTEM

MEMORY:

The size of the memory used in the designing of the system is 4096 × 16. It means that the memory consists of 4096 words, each word of 16 bits. Since the memory consists of 16 bits, it requires 16 separate connection lines. We have assumed the stored program concept so that the instructions to be executed are stored in the memory where AR determines the address of the instructions through separate address lines. These instructions hold the operand or address of the operand on which the operation is done.

Two input lines read and write are used to read from and write into the memory. In this way the memory read/write operation is done in the system.

ADDRESS REGISTER (AR):

It is a 12-bit register which holds address for the memory. The address register consists of separate address line to the memory due to which data addressing for all the registers is carried out efficiently and economically by the reduction of requirement of separate address lines for each register.

INSTRUCTION REGISTER (IR):

It holds a 16-bit instruction code which is pointed by the address loaded from PC to AR. The first 12 bits, 0 to 11, is the address of the operand whereas the remaining 4 bits, from 12 to 15, is the opcode (operation code) that determines the task to be performed with the operand. The opcode bit, from bit value 12 to14, is decode by using a 3×8 decoder giving the output D_0-D_7.

PROGRAM COUNTER (PC):

A Program Counter is a register which holds the address of the next instruction to be executed. It is a 12-bit register which gives the value of only the address part to AR.

DATA REGISTER (DR):

A Data Register is a 16-bit register which holds the value of the operand on which the operation is to be done. It fetches the value of the memory address pointed by AR.

ACCUMULATOR (AC):

It is a single processor register which functions as the next destination register for the value where the operation is to be done. It is a 16-bit register.

INPUT (INPR)/OUTPUT REGISTER (OUTR):

Two registers, namely input register (INPR) and output register (OUTR), are used to transfer the data from and to peripheral devices. They have an important purpose as the transfer has to be done from and to the computer. Both of the register transfer 8-bit data from and to the input and output devices.

TEMPORARY REGISTER (TR):

It acts as resting place for the data for some time until the instruction is executed. After the execution of the instruction register transfer the data to the destination.

ARITHMETIC AND LOGIC UNIT (ALU):
It is the heart of the computer system design. It performs the various operations on the data from accumulator and the data provided by data register. The operations performed by ALU are listed below:

- ADD
- AND
- SHIFT RIGHT
- SHIFT LEFT
- OR
- COMPLEMENT ACCUMULATOR

CONTROL UNIT:
It is the combination of the timing signal and the decoded output from the instruction register which is defined for the certain interval of time. In this interval of time, a certain micro operation is done which is controlled by the control unit. In this way the control input helps to perform different functions in the ALU.

OPCODE:
An Opcode is a set of certain bits allocated to perform a certain type of instruction stored in the memory.
The different types of instructions are defined by the value of the opcode and the most significant bit of the value of the instruction register.
The different types of instructions defined by the opcode are as follows:

MEMORY REFERENCE INSTRUCTION:
It is differentiated from the other by the value of opcode which is shown below:

I=1	**Opcode**	**Address**

Fig: Memory reference format

REGISTER REFERENCE INSTRUCTION:

0	**1 1 1**	**RRI Opcode**

Fig: Register Reference Format

INPUT/OUTPUT REFERENCE:

1	**1 1 1**	**RRI** **Opcode**

Fig: input/output Reference

INSTRUCTION SETS AND FLOW DIAGRAM

The different instruction sets used for the proper running of the computer system with the flow chart of the computer cycle is as follows:

Symbol	Binary Code	Description
Memory Reference Instructions (MRI)		
AND	000	And memory word to AC
ADD	001	Add memory word to AC
LDA	010	Load memory word to AC
STA	011	Store data of AC in memory address
BUN	100	Branch unconditionally to AC
BSA	101	Branch and store return address
ISZ	110	Increment and skip if zero
Register Reference Instructions (RRI)		
CLA	0111 1011	Clear data in AC
CLE	0111 1010	Clear E
CMA	0111 1001	Complement AC
CME	0111 1000	Complement E
CIR	0111 0111	Circulate right AC and E
CIL	0111 0110	Circulate left AC and E
INC	0111 0101	Increment AC
SPA	0111 0100	Skip next instruction if AC is positive
SNA	0111 0011	Skip next instruction if AC is negative
SZA	0111 0010	Skip next instruction if AC is 0
SZE	0111 0001	Skip next instruction if E is 0
HLT	0111 0000	Halt computer
Input Output Instructions		
INP	1111 1011	Input character to AC
OUT	1111 1010	Output character to AC
SKI	1111 1001	Skip on input flag
SK0	1111 1000	Skip on output flag
ION	1111 0111	Interrupt On
IOF	1111 0110	Interrupt Off

FLOW DIAGRAM OF SYSTEM:

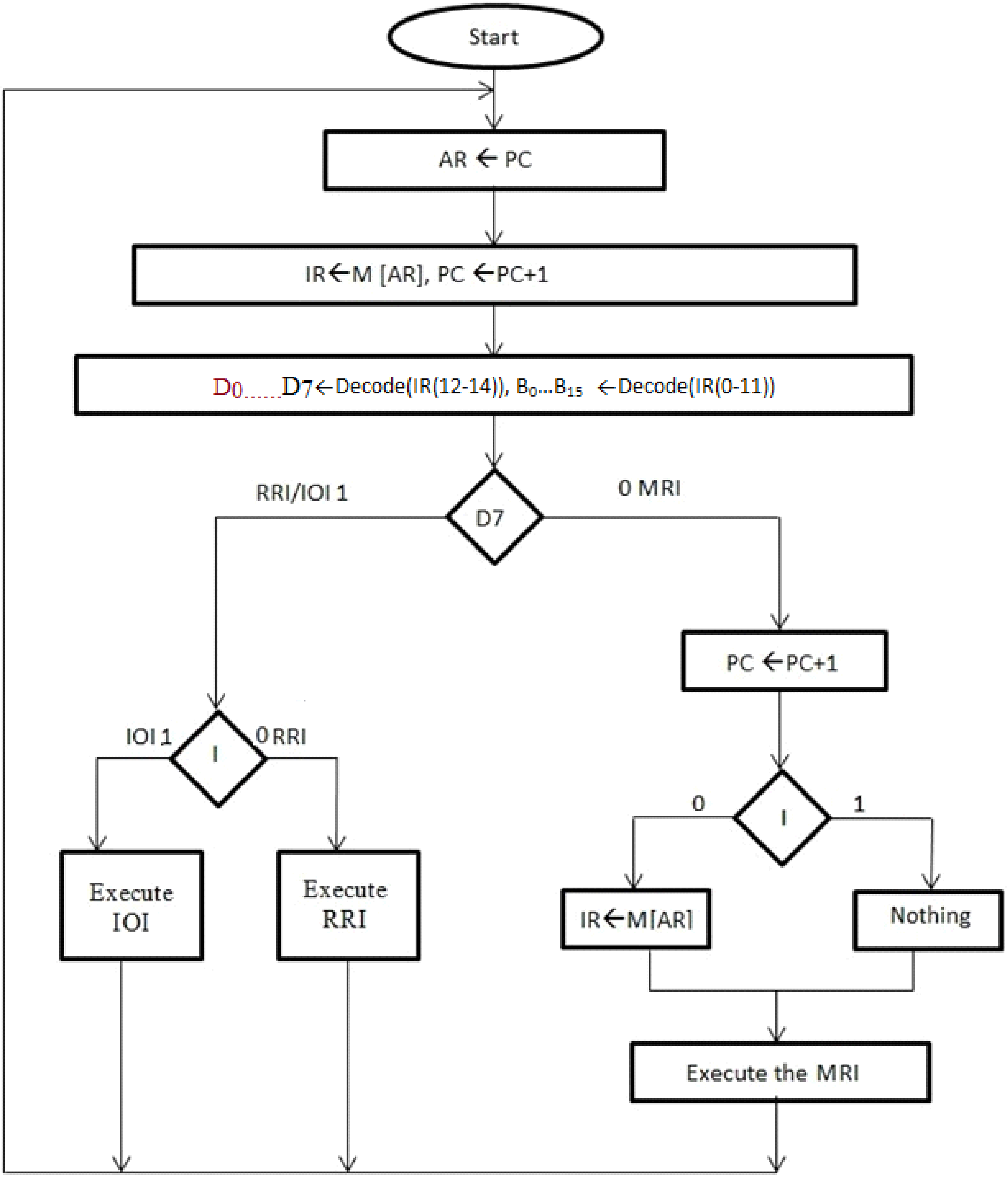


Fig: Flow diagram of the system

EXPLANATION OF FLOW DIAGRAM AND INSTRUCTION SETS

FETCH AND DECODE CYCLE:

In this cycle, the instruction from PC is send to the AR. The data value fetched by AR is send to IR through memory write line enabled. Then, the received value is decoded from there to check whether it is a register reference or memory reference or input/output instruction. In this way fetch and decode cycle is completed.

The fetch and decode cycle in terms of the RTL is shown below:

Cycle	Control Signal	RTL	Description
Fetch	**T_0**	**AR ← PC**	Load AR with the address of new instruction to be read from the memory
	T_1	**IR ← M[AR], PC ← PC + 1, AR ← AR + 1**	Copy the instruction read from the memory to the program counter.
Decode	**T_2**	**D_0 … D_7 ← Decode[IR(12-14)], I ← IR(15), AR ← M[AR]**	Decode the Instruction

After the fetch and decode cycle, the value of the decoded input is checked whether it is of memory or register or input /output; then the execution phase is initiated.

EXECUTION CYCLE:

In the execution cycle, different instruction sets performs the different micro operation which is described below:

ADD:

In this micro operation the data value from the accumulator and data register is done logical OR which is shown in terms of RTL as follows:

$$D_0T_4: \quad DR \leftarrow M\,[AR]$$
$$D_0T_5: \quad AC \leftarrow AC+DR,\ E \leftarrow C_{out},\ SC \leftarrow 0$$

LDA:

This control function enables the transfer of content from the memory specified place to accumulator. The RTL of instruction is as follows:

$$D_1T_4: \quad DR \leftarrow M\,[AR]$$
$$D_1T_5: \quad AC \leftarrow DR,\ SC \leftarrow 0$$

AND:

This performs the logical AND-ing operation between the accumulator value and the data register value specified by the memory. The RTL is shown below:

D_2T_4: DR ← M [AR]
D_2T_5 AC ← AC^DR, SC ← **0**

BSA:

This instruction is especially helpful in implementing subroutines calls. It causes the computer to branch to memory location specified by the address in the meantime save the current address of the PC so that after the completion of the CALL, the program can continue its normal execution. During the first cycle the content of the program counter is saved in the memory location pointer by the AR. AR is also increment by one. On the next execution state, the address stored in the AR is copied to the PC and SC is reset. RTL for this instruction is given below:

D_3T_4: M [AR] ← PC, AR ← AR+1
D_3T_5: PC ←AR, SC ← 0

BUN:

This instruction causes the computer to branch unconditionally to the given address and continue the execution of instruction for the location. It takes only one step to execute this instruction. To execute this instruction content of the address is simply transferred to the program counter and the sequence counter is reset.

D_4T_4: PC ← AR, SC ← 0

ISZ:

Increment and skip on zero instruction increments the content of the memory location specified by the address and if the result after incrementing is zero then it skips the next instructions. In the first timing signal, the content of the memory location pointed by the address is copied to the data register, on the next cycle data register is incremented by one, finally on the last timing signal the content is copied back to the same memory location and if the content is zero then the computer skips the next instruction in the memory by incrementing the program counter by one.

D_6T_4: DR←M [AR]
D_6T_5: DR←DR+1
D_6T_6: M [AR] ←DR, IF (DR←0) THEN (PC←PC+1), SC←0

REGISTER REFERENCE INSTRUCTION:

CLA:

This is instruction helps to clear the accumulator value to zero.

RTL:

rB_0: AC ← 0

HLT:

It start-stops the whole computer system by enabling the value of the IEN register.

RTL:

rB3: S ← 0

CIR:

This is the circular right shift instruction and causes the content of the accumulator to be circularly shifted right without loss of any bit. This operation is carried out in ALU.

RTL:

rB1: AC ← Shr AC, AC (15) ← E, E ← AC (0)

CIL:

This is the circular left shift instruction and causes the content of the accumulator to be circularly shifted left without loss of any bit. This operation is carried out in ALU.

RTL:

rB_2: AC ← Shl AC, AC (0) ← E, E ← AC (15)

CMA:

The content of the AC is complemented by this instruction. The content of the AC is passed though the ALU where the complement operation is select, finally the complemented result is stored back in the ALU.

INPUT/OUTPUT INSTRUCTION:

INP:

This is input instruction which enables the transfer the information from the input device to the accumulator.

RTL:

pB_{11}: AC (0-7) ← INPR, FGI ← 0

OUT:

It is the output instruction which helps to take out the instruction from the accumulator to output device.

RTL:

pB_{10}: OUTR ← AC (0-7), FGO ← 0

ION

This instruction set an interrupt enable flip-flop IEN i.e. interrupt enable on.

pB_7: IEN ← 1

IOF

This instruction clear an interrupt enable flip-flop IEN i.e. Interrupt enable off.

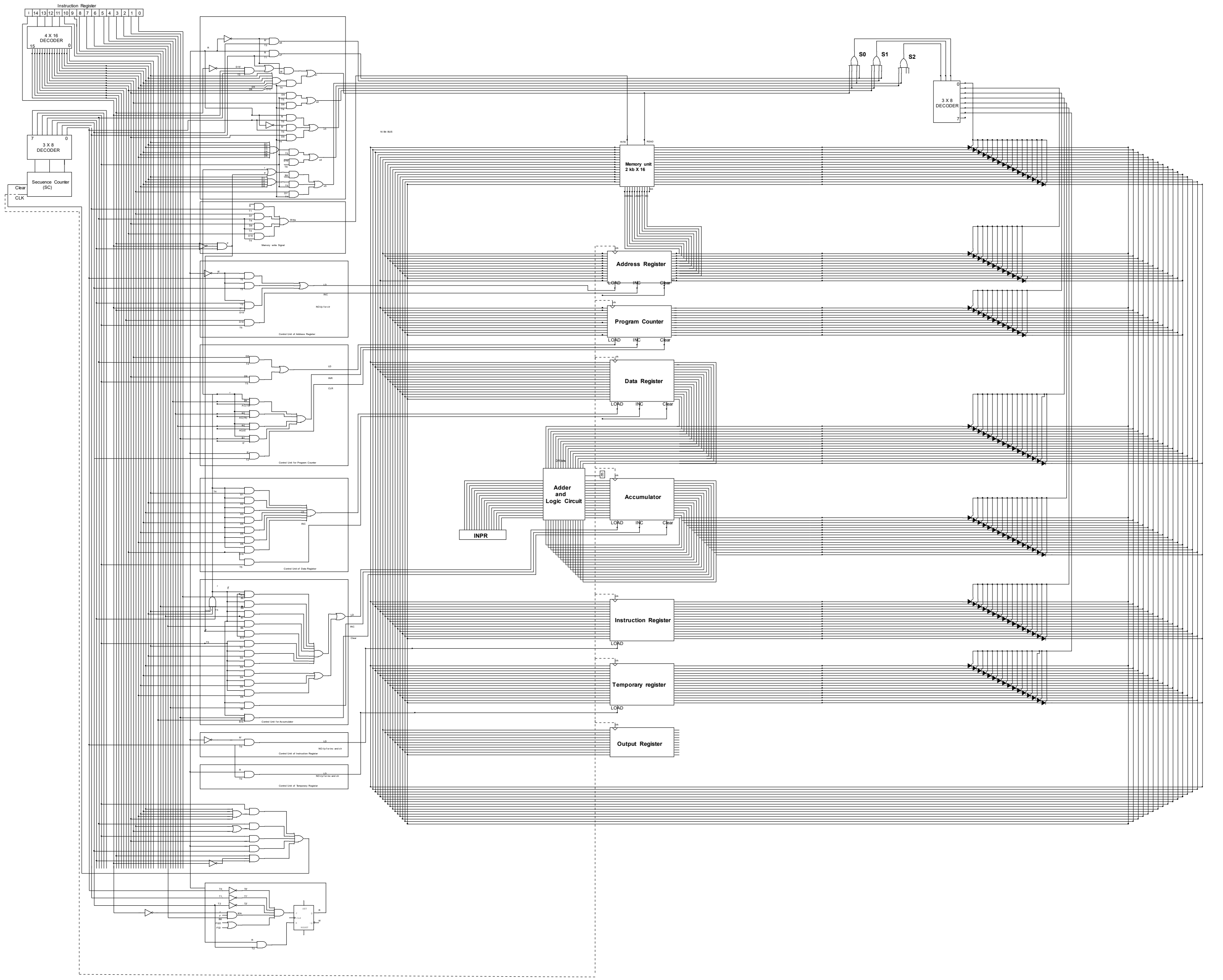
pB_6: IEN ← 0

CONCLUSION

This project provides the basic knowledge necessary to understand hardware operation of digital computer. It imparts clear vision on how the register transfer micro-operation works in a computer and also provides concept on the working mechanism of the ALU, register, multiplexer and counter. I would also like to thank to course instructor Mr. Dhiraj Shrestha for providing an opportunity to implement the theoretical concept learned in Computer architecture organization via mini-project to design 16 bit computer.

REFERENCE

- M. MORRIS MANO. COMPUTER SYSTEM. ARCHITECTURE. Third Edition



Title: 16-Bit CPU		
16 bit computer		
Designed by:Manish Karn	Document N:1	Revision: 1
Checked by:	Date: 8/2/2015	Size: A0
Approved by:	Sheet 1 of 1	