DRONACHARYA
Group of Institutions

B-27, Knowledge Park – III, Greater Noida Uttar Pradesh - 201308
Approved by: All India Council for Technical Education (AICTE), New Delhi
Affiliated to: Dr. A. P. J. Abdul Kalam Technical University (AKTU), Lucknow

DEPARTMENT OF INFORMATION TECHNOLOGY

Computer Organization and Architecture

Lab Manual

SUBJECT CODE: BCS-352

B.Tech., Semester -III

Session: 2024-25, ODD Semester

Table of Contents

DRONACHARYA GROUP OF INSTITUTIONS GREATER NOIDA

VISION

- Instilling core human values and facilitating competence to address global challenges by providing Quality Technical Education.

MISSION

- M1 - Enhancing technical expertise through innovative research and education, fostering creativity and excellence in problem-solving.
- M2 - Cultivating a culture of ethical innovation and user-focused design, ensuring technological progress enhances the well-being of society.
- M3 - Equipping individuals with the technical skills and ethical values to lead and innovate responsibly in an ever-evolving digital land.

DEPARTMENT OF INFORMATION TECHNOLOGY

VISION

To provide students with theoretical understanding and technical proficiency in Information Technology, instill with moral and ethical values, to excel in academic, industry, and research settings.

MISSION

M1: To instill in students a strong foundation of both the theory and practical application of IT skills, combined with the innovation and research approaches to keep pace with emerging technologies.

M2: To empower graduates to become global leaders specializing in field of Information Technology.

M3: To impart to students the social, ethical, and moral values necessary for them to make substantial contributions to society.

Program Outcomes (POs)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.

PO 9: Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply theseto one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Programme Educational Objectives (PEOs)

PEO1: Apply the knowledge of mathematics, science and engineering fundamentals to identify and solve IT and engineering problems.

PEO2: Use various software tools and technologies to solve problems related to academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

Program Specific Outcomes (PSOs)

PSO1: Ability to think logically and apply programming knowledge and practices in analyzing real world problems and provide solutions to meet the needs of society.

PSO2: Enhance the competence of technocrats to provide professional engineering solutions as per the industrial and societal needs.

PSO3: To cultivate and produce highly motivated engineers committed to lifelong learning, pursuing research, higher education, and embracing competitive challenges to excel in the future.

University Syllabus

1. Implementing HALF ADDER,FULL ADDER using basic logic gates
2. Implementing Binary-to-Gray,Gray-to-Binary code conversions.
3. Implementing 3-8 line DECODER.
4. Implementing 4x1and 8x1 MULTIPLEXERS.
5. Verify the excitation tables of various FLIP-FLOPS.
6. Design of an 8-bit Input/Output system with four8-bit Internal Registers.
7. Design of an 8-bit ARITHMETIC LOGIC UNIT.
8. Design the data path of a computer from its register transfer language description.
9. Design the control unit of a computer using either hard wiring or micro programming based on its register transfer language description.
10. Implement a simple instruction set computer with a control unit and a data path.

Course Outcomes(COs)

Upon successful completion of the course, the students will be able to

C.1: Design and verify combinational circuits (adder, code converter, decoder, multiplexer) using basic gates.
C.2: Design and verify various flip-flops.
C.3: Design I/O system and ALU.
C.4: Demonstrate a simple instruction set computer.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
C.1	3		3					2	2	2		2
C.2	3		3					2	2	2		2
C.3	3		3					2	2	2		2
C.4	2		3					2	2	2		

CO-PSO Mapping

	PSO1	PSO2	PSO3
C207.1	-	3	-
C207.2	-	3	-
C207.3	-	3	-
C207.4	-	3	-
C207	**-**	**3**	**-**

Course Overview

An important part of the undergraduate curriculum for IT students is coverage of computer organization. Typically, this is accomplished by both a lecture and lab course. The purpose of thelabcourseistohavethestudentsdeveloppracticaldesignskills.Ourcomputerorganization laboratory course starts with traditional logic gate design, then gradually incorporates control unit, up to the point where students are building a simple instruction set computer. The laboratory course is scheduled for a two-hour time slot. There are ten lab experiments.

During the first set of Experiments, students become familiar with basic digital hardware by constructing simple combinational circuits, and learning troubleshooting skills. During the secondsetofExperiments,studentsbecomefamiliarwitharithmeticlogicunitandcontrolunit.
Finally, students apply their knowledge to the design a simple instruction set computer. Because students are constructing complete computer organization projects, the computer organization lab typically requires more effort than traditional laboratory courses.

ListofExperimentsmappedwithCOs

SLno.	*ListofExperiments*	*Course Outcome*
1.	Implementing HALF ADDER,FULL ADDER using basic logic gates	CO.1
2.	Implementing Binary-to-Gray,Gray-to-Binary code conversions.	CO.1
3.	Implementing 3-8 line DECODER**.**	CO.1
4.	Implementing 4x1 and 8x1 MULTIPLEXERS.	CO.1
5.	Verify the excitation tables of various FLIP-FLOPS.	CO.2
6.	Design of an 8-bit Input/Output system with four8-bit Internal Registers.	CO.2
7.	Design of an 8-bit ARITHMETIC LOGIC UNIT.	CO.3
8.	Design the data path of a computer from its register transfer language description.	CO.4
9.	Design the control unit of a computer using either hardwiring or micro programming based on its register transfer language description.	CO.4
10.	Implement a simple instruction set computer with a control unit and a data path.	CO.4

Dos and DON'Ts

Dos

1. Login-on with your user name and password.
2. Log off the Computer every time when you leave the Lab.
3. Arrange your chair properly when you are leaving the lab.
4. Put your bags in the designated area.
5. Ask permission to print.

DON'Ts

1. Do not share your user name and password.
2. Do not remove or disconnect cables or hardware parts.
3. Do not personalize the computer setting.
4. Do not run programs that continue to execute after you log off.
5. Do not download or install any programs, games or music on computer in Lab.
6. Personal Internet use chat room for Instant Messaging (IM) and Sites Strictly Prohibited.
7. No Internet gaming activities allowed.
8. Tea, Coffee, Water & Eatables are not allowed in the Computer Lab.

General Safety Precautions

Precaution (In case of Injury or Electric Shock)

1. To break the victim with live electric source .Use an insulator such as fire wood or plastic to break the contact. Do not touch the victim with bare hands to avoid the risk of electrifying yourself.
2. Unplug the risk of faulty equipment. If main circuit breaker is accessible, turn the circuit off.
3. If the victim is unconscious, start resuscitation immediately, use your hands to press the chest in and out to continue breathing function. Use mouth-to-mouth resuscitation if necessary.
4. Immediately call medical emergency and security. Remember! Time is critical; be best.

Precaution (In case of Fire)

1. Turn the equipment off. If power switch is not immediately accessible, take plug off.
2. If fire continues, try to curb the fire if possible by using the fire extinguisher or by covering it with a heavy cloth if possible isolate the burning equipment from the other surrounding equipment.
3. Sound the fire alarm by activating the nearest alarm switch located in the hall way.
4. Call security and emergency department immediately:

Guidelinestostudentsforreportpreparation

All students are required tomaintain a record of the experiments conducted bythem. Guidelines for its preparation are as follows: -

1) Allfilesmustcontainatitlepagefollowedbyanindexpage.***Thefileswillnotbesignedby the faculty without an entry in the index page.***

2) Student'sName,Rollnumberanddateofconductionofexperimentmustbewrittenonall pages.

3) Foreachexperiment,therecordmustcontainthefollowing

(i) Aim/Objectiveoftheexperiment

(ii) Equipment'srequired

(iii) Pre-experimentwork(as givenbythefaculty)

(iv) Observationtable

(v) Results/output

Note:

1. Studentsmustbringtheirlabrecordalongwiththemwhenevertheycomeforthe lab.

2. Studentsmustensurethattheirlabrecordisregularlyevaluated.

LabAssessmentCriteria

Anestimated10labclasses areconducted inasemesterfor eachlabcourse. Theselabclasses are assessed continuously. Each lab experiment is evaluated based on 5 assessment criteria as shown in following table. Assessed performance in each experiment is used to compute CO attainment as well as internal marks in the lab course.

Grading Criteria	**Exemplary(4)**	**Competent(3)**	**Needs Improvement (2)**	**Poor(1)**
AC1: Pre-Lab written work (for last lab class, this may be assessed through viva)	Complete procedure with underlined concept is properly written	Underlined concept is written but procedure is incomplete	Notabletowrite concept and procedure	Underlined concept is not clearly understood
AC2: Program Writing/ Modeling	Assignedproblem is properlyanalyzed, correct solutiondesigned, appropriate language constructs/ tools are applied, Program/solution written is readable	Assigned problem is properly analyzed, correctsolution designed, appropriate language constructs/tools are applied	Assigned problem is properly analyzed & correct solution designed	Assigned problem is properly analyzed
AC3: Identification & Removalof errors/ bugs	Able to identify errors/ bugs and remove them	Abletoidentify errors/ bugsand remove them with little bit of guidance	Is dependent totally on someone for identification of errors/ bugs and their removal	Unable to understand the reasonfor errors/ bugs evenafterthey are explicitly pointed out
AC4:Execution &Demonstration	All variants of input /output are tested,Solutionis welldemonstrated and implemented concept is clearly explained	Allvariantsof input /output are not tested, However, solution is well demonstrated and implemented concept is	Only few variantsofinput /output are tested, Solutioniswell demonstratedbut implemented concept is not clearlyexplained	Solution isnot well demonstrated and implemented concept is not clearly explained

		clearly explained		
AC5:Lab Record Assessment	All assigned problemsarewell recorded with objective,design constructsand solution along with Performance analysisusingall variantsofinput and output	Morethan70% of the assigned problems are well recorded with objective, designcontracts and solution along with Performance analysisisdone with all variants of input and output	Lessthan70% of the assigned problems are well recorded with objective, design contracts and solution along with Performance analysisisdone with all variants of input and output	Lessthan40% of the assigned problems are well recorded with objective, designcontracts and solution along with Performance analysisisdone with all variants of input and output

LABEXPERIMENTS

LABEXPERIMENT1

OBJECTIVE:DesignandimplementationofHalfAdderandFullAdder.

EQUIPMENTS&COMPONENTSREQUIRED:

SL.No.	Equipment's	Specification	Quantity
1	DigitalICTrainerkit	-	1
2	DigitalMultimeter		1

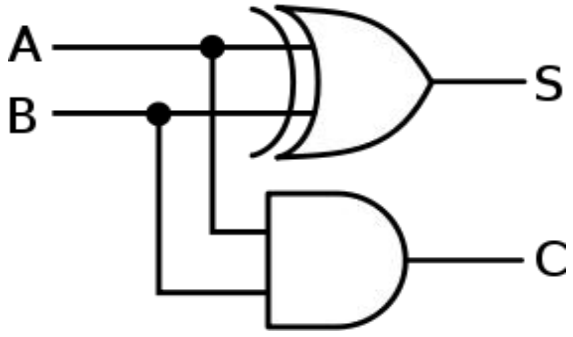
SL.No.	Components	Specification	Quantity
1	DigitalICs	7400,7402,7404, 7408,7432,7486.	1each
2	Patchcords	-	6

BRIEFDESCRIPTION:

- Todesignandimplementhalfadderusinglogic gates

HALFADDER

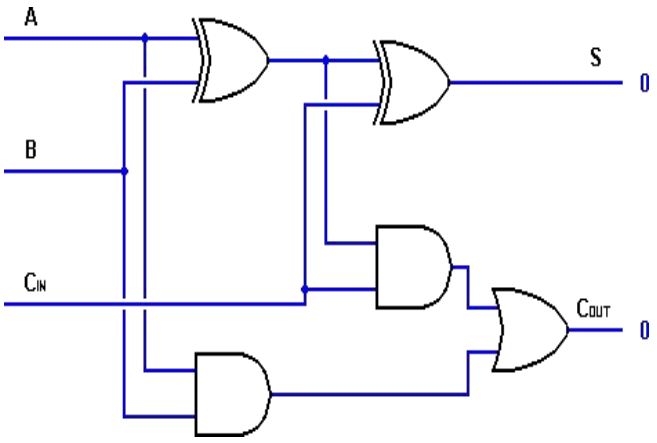
INPUTA	INPUTB	OUTPUTS	
		S	C
0	**0**	**0**	**0**
0	**1**	**1**	**0**
1	**0**	**1**	**0**
1	**1**	**0**	**1**



CircuitDiagram TruthTable

- Todesignandimplementfulladderusinglogic gates.

FULLADDER



Input bit for number A	Input bit for number B	Carry bit input C_{IN}	Sum bit output S	Carry bit output C_{OUT}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

CIRCUITDIAGRAM **TRUTHTABLE**

PRE-EXPERIMENTQUESTIONS:-

1. Explainthetruthtableofhalfadder.
2. HowmanyEx-orandor orgatecanbeusedtomakeahalfadder?

PROCEDURE:-

- Identifythepins.
- Connectthecircuitaspercircuitdiagram.
- Obtainoutputswithvariousinputcombinations.
- VerifyitwiththeBooleanfunctionusingtruthtable

POST-EXPERIMENTQUESTIONS:-

1. Whataretheapplicationsofhalfadder?
2. Whataretheapplicationoffulladder?

LABEXPERIMENT2

OBJECTIVE:DesignandimplementationofBinarytoGray,GraytoBinaryCode conversions

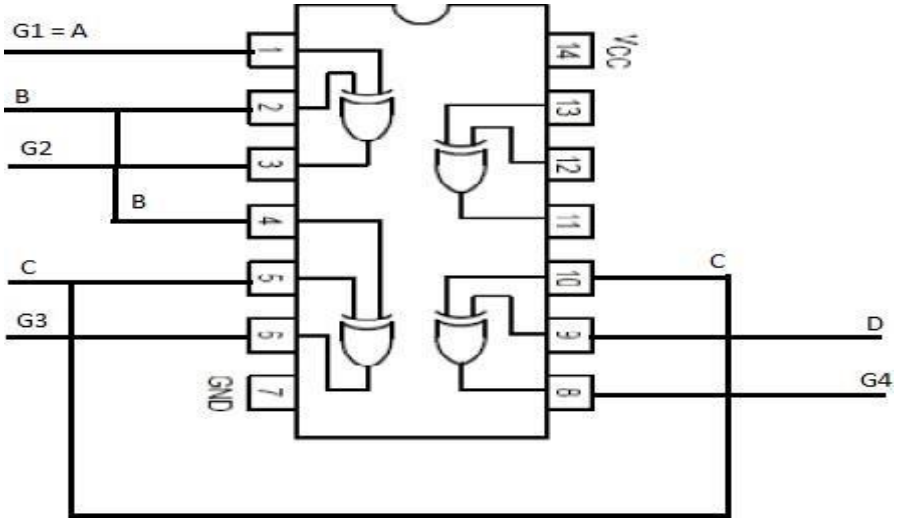
EQUIPMENTS&COMPONENTSREQUIRED:

SL. No.	Equipments	Specification	Quantity
1	DigitalICTrainerkit	-	1
2	DigitalMultimeter		1

SL. No.	Components	Specification	Quantity
1	DigitalICs	7400,7402, 7404, 7408,7432, 7486.	1each
2	Patchcords	-	6

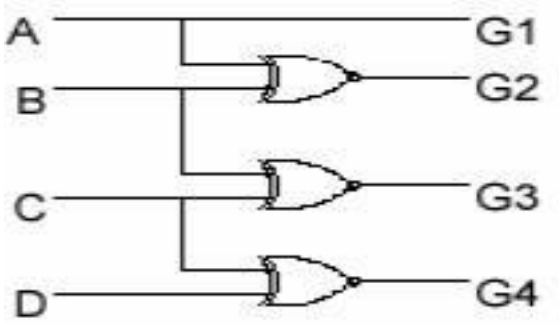
BRIEFDESCRIPTION:

a) TodesignandimplementBinarytoGrayCodeconversions



PindiagramofBinarytograycodeconverterusing7486IC(ex-or Gate)

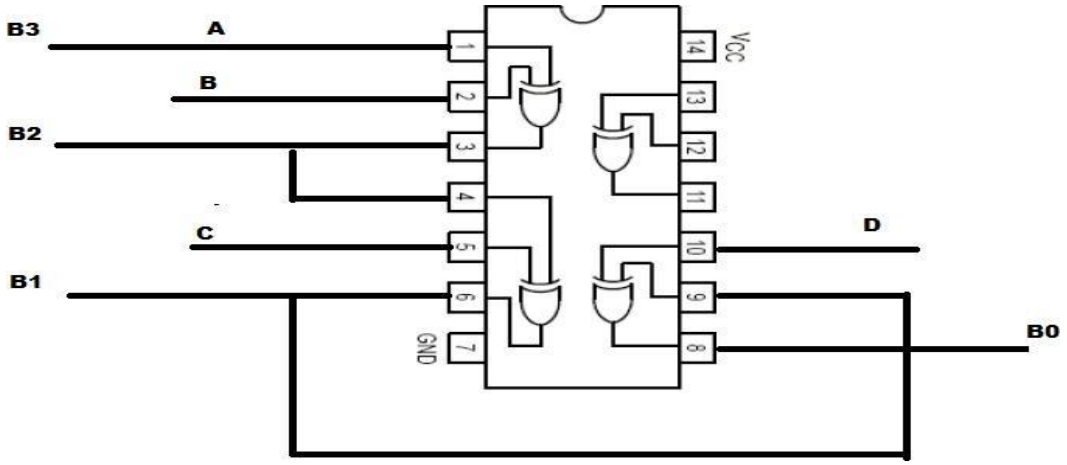
INPUTS				OUTPUTS			
A	B	C	D	G_4	G_3	G_2	G_1
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0



CircuitDiagramofBinarytoGrayCode Converter

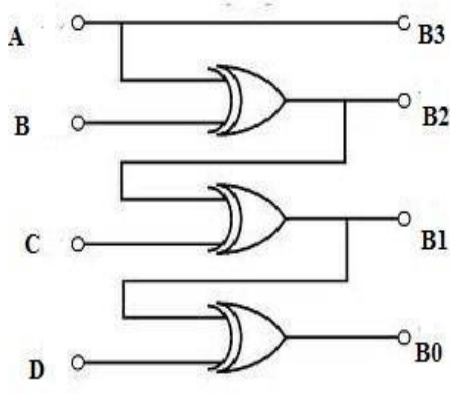
TruthTable

b) TodesignandimplementBinarytoGrayCodeconversions



PindiagramofGraytoBinarycodeconverter using7486Ic(Exor Gate)

INPUTS				OUTPUTS			
A	**B**	**C**	**D**	**B_3**	**B_2**	**B_1**	**B_0**
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1



CircuitDiagramforGraytoBinaryCodeConverter

TRUTHTABLE

PRE-EXPERIMENTQUESTIONS:-

1 Whatisacodeconverter.

2 Differentiatebetweentranslatorandcodeconverter.

PROCEDURE:-

- Collectthecomponentsnecessarytoaccomplishthisexperiment.
- PlugtheICchipintothebreadboard.
- Connectthesupplyvoltageandgroundlinestothechips.PIN7=GroundandPIN14= +5V.
- Makeconnectionsasshownintherespectivecircuitdiagram.
- Connecttheinputsofthegatetotheinputswitches oftheLED.
- ConnecttheoutputofthegatetotheoutputLEDs.
- Onceallconnectionshavebeendone,turnonthepowerswitchofthebreadboard
- Operatetheswitchesand fillinthetruthtable(Write"1"if LEDisONand "0"ifL1 is OFF Apply the various combination of inputs according to the truth table and observe the condition of Output LEDs.

POST-EXPERIMENTQUESTIONS:-

1. Whataretheadvantagesofcodeconverter?
2. Whatarethepropertiesofgraycode?

LABEXPERIMENT3

OBJECTIVE: Designandimplementationof2-4and3-8linedecoder.

EQUIPMENTS&COMPONENTSREQUIRED:

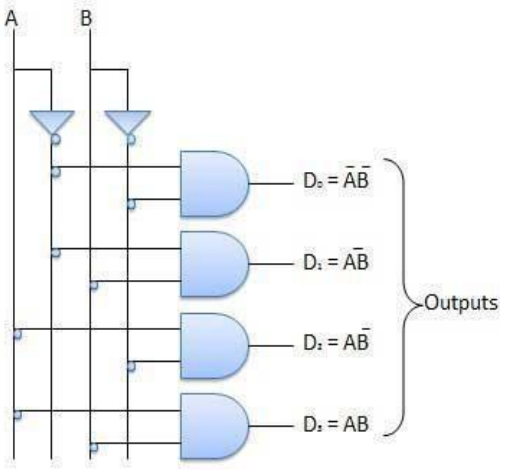
SL.No.	Equipments	Specification	Quantity
1	DigitalICTrainerkit	-	1
2	DigitalMultimeter		1

S	Components	Specification	Quantity
1	DigitalICs	7400,7402, 7404, 7408,7432, 7486.	1each
2	Patchcords	-	6

BRIEFDESCRIPTION:

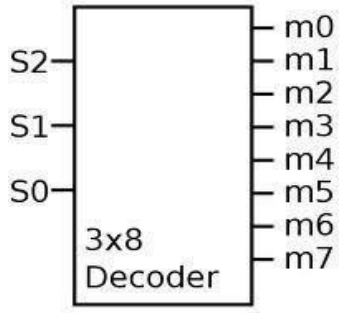
a) 2to4Decoderusinglogicgates:

$\overline{E}$	A_1	A_0	$\overline{D}_0$	$\overline{D}_1$	$\overline{D}_2$	$\overline{D}_3$
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0
1	X	X	1	1	1	1



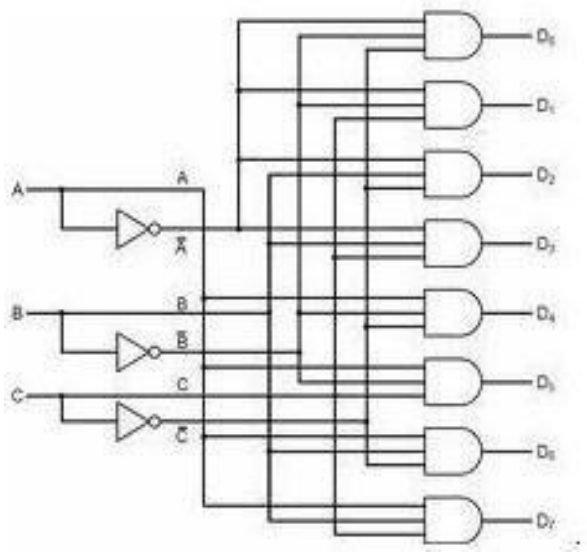
TruthTable **LogicDiagram**

b) 3to8decoderusinglogic gates:



A	B	C	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

SYMBOL **TRUTHTABLE**



LOGICDIAGRAMOF3TO8DECODER:

PRE-EXPERIMENTQUESTIONS:

1. DifferencebetweenEncoderandDecoder.
2. Explaintheneedofmultiplexer.

PROCEDURE:

- Collectthecomponentsnecessarytoaccomplishthisexperiment.
- PlugtheICchipintothebreadboard.
- Connectthesupplyvoltage andgroundlinestothechips.PIN7=Ground
- andPIN14=+5V.
- Makeconnectionsasshownintherespectivecircuitdiagram.
- Connecttheinputsofthegatetotheinputswitches oftheLED.
- ConnecttheoutputofthegatetotheoutputLEDs.
- Onceallconnectionshavebeendone,turnonthepowerswitchofthebreadboard

- Operatetheswitchesandfillinthetruthtable(Write"1"ifLEDisONand"0"ifL1 is OFF Apply the various combination of inputs according to the truth tab alend observe the condition of Output LEDs.

POSTEXPERIMENTQUESTIONS:

1. Designa5to32decoderusingone2to4and four3 to8decoderic's.
2. Writeanoteon BCDtodecimal decoder.

LABEXPERIMENT4

OBJECTIVE: ImplementationOf4x1And8x1Multiplxer

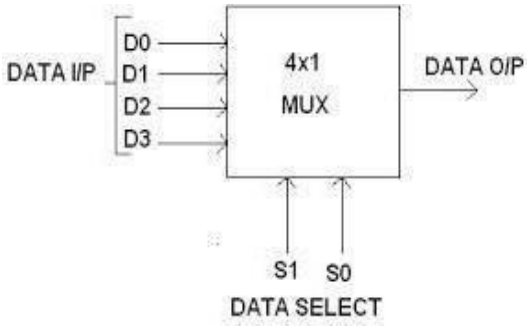
EQUIPMENTS&COMPONENTSREQUIRED:

SL.No.	Equipment's	Specification	Quantity
1	DigitalICTrainerkit	-	1
2	DigitalMultimeter		1

SL.No.	Components	Specification	Quantity
1	DigitalICs	7400,7402, 7404, 7408,7432, 7486.	1each
2	Patchcords	-	6

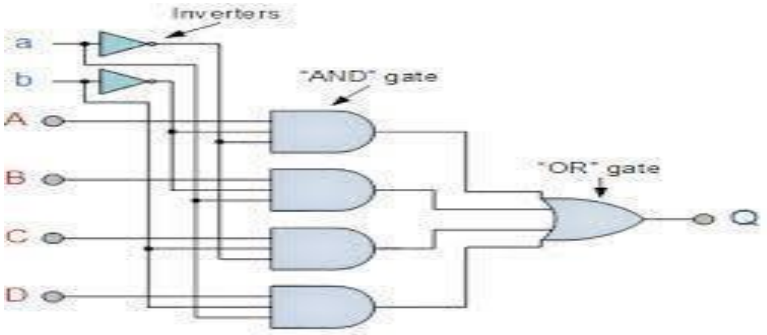
BRIEFDESCRIPTION:

a)4to1MULTIPLEXERS:



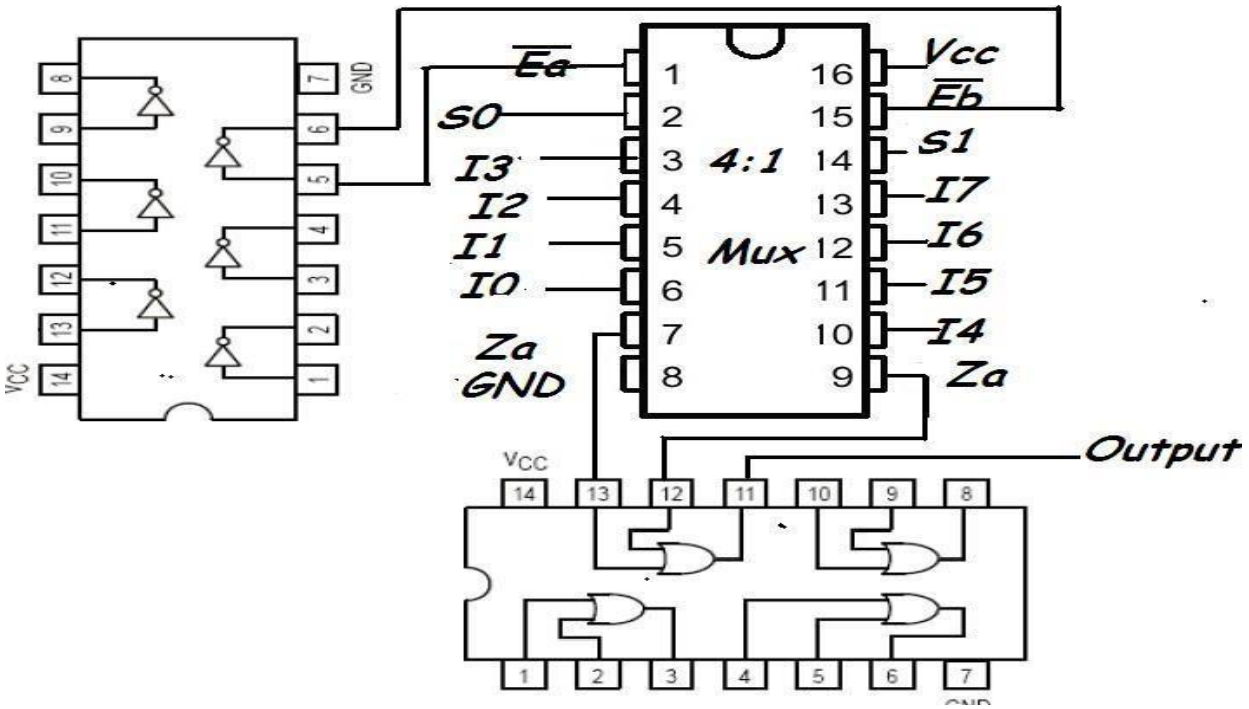
Addressing		Input Selected
b	a	
0	0	A
0	1	B
1	0	C
1	1	D

LOGICDIAGRAM: **TRUTHTABLE:**

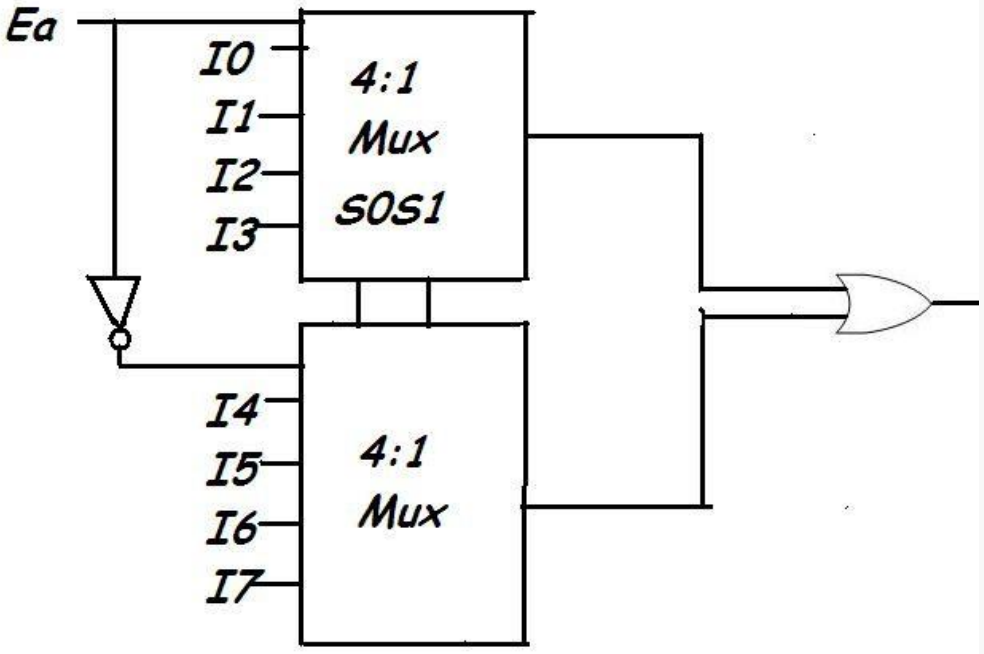


$$Q=\bar{a}\bar{b}A+\bar{a}bB+a\bar{b}C+abD$$

d) 8x1Multiplexer



PINDIAGRAMOF8:1MUXUSINGTWO4:1MUX



CIRCUITOF8:1MUXUSINGDUAL4:1MUX

SelectLines			Inputs								Output		
E_a	S_0	S_1	I_o	I_1	I_2	I_3	I_4	I_5	I_6	I_7	Z_a	Z_b	Y
0	0	0	0	×	×	×	×	×	×	×	0	×	0
0	0	0	1	×	×	×	×	×	×	×	1	×	1
0	0	1	×	0	×	×	×	×	×	×	0	×	0
0	0	1	×	1	×	×	×	×	×	×	1	×	1
0	1	0	×	×	0	×	×	×	×	×	0	×	0
0	1	0	×	×	1	×	×	×	×	×	1	×	1
0	1	1	×	×	×	0	×	×	×	×	0	×	0
0	1	1	×	×	×	1	×	×	×	×	1	×	1
1	0	0		×	×	×	0	×	×	×	×	0	0
1	0	0	×	×	×	×	1	×	×	×	×	1	1
1	0	1	×	×	×	×	×	0	×	×	×	0	0
1	0	1	×	×	×	×	×	1	×	×	×	1	1
1	1	0	×	×	×	×	×	×	0	×	×	0	0
1	1	0	×	×	×	×	×	×	1	×	×	1	1
1	1	1	×	×	×	×	×	×	×	0	×	0	0
1	1	1	×	×	×	×	×	×	×	1	×	1	1

TRUTHTABLEOF8:1MUXUSINGDUAL4:1MUX PRE

EXPERIMENT QUESTIONS:-

1. DifferencebetweenEncoderandDecoder.
2. Explaintheneedofmultiplexer.

PROCEDURE:-

- Collectthecomponentsnecessarytoaccomplishthisexperiment.
- PlugtheICchipintothebreadboard.
- Connectthesupplyvoltageandgroundlinestothechips.PIN7=Ground and PIN14 = +5V.
- Makeconnectionsasshownintherespectivecircuitdiagram.

- Connecttheinputsofthegatetotheinputswitches ofthe LED.
- ConnecttheoutputofthegatetotheoutputLEDs.
- Onceallconnectionshavebeendone,turnonthepowerswitchofthebreadboard
- Operatetheswitchesandfillinthetruthtable(Write"1"ifLEDisONand"0"ifL1 is OFF Apply the various combination of inputs according to the truth table and observe the condition of Output LEDs.

POSTEXPERIMENTQUESTION:-

1. Designa16 to1 Multiplexerusingone4to1MuxIC'S.

LABEXPERIMENT5

OBJECTIVE: VerifytheexcitationtablesofvariousFLIP-FLOPS.

EQUIPMENTS&COMPONENTSREQUIRED:

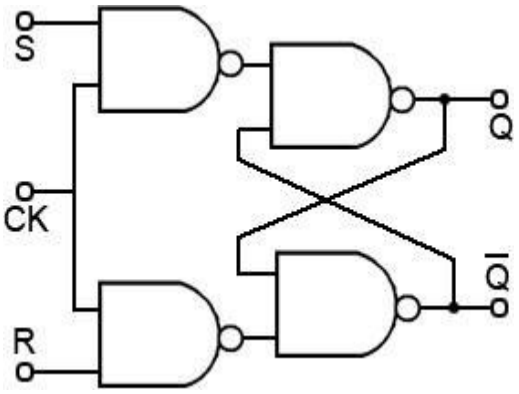
DigitalICtrainerkit,IC7400,IC7410,IC7473,IC7474,IC 7476

BRIEFDESCRIPTION:

Flipflopsarethebasicbuildingblocksinanymemorysystemssinceitsoutputwill remain in its state until it is forced to change it by some means.

SRFLIP FLOP:

SandRstandsforsetandreset.Therearefourinputcombinationpossibleattheinputs. But = =1 is forbidden since the output will be indeterminate.

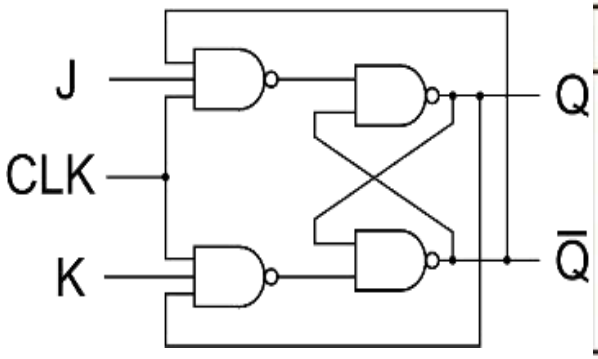


INPUTS			OUTPUT	STATE
CLK	S	R	Q	
X	0	0	No Change	Previous
↑	0	1	0	Reset
↑	1	0	1	Set
↑	1	1	-	Forbidden

LOGIC DIAGRAM OF R S FLIP FLOPS: **TRUTHTABLE:**

J K FLIP FLOP

TheindeterminateoutputstateofSRFFwhenS=R=1isavoidedbyconvertingittoaJKFF. When flip flop is switched on its output state is uncertain. When an initial state is to be assigned two separate inputs called preset and clear are used. They are active low inputs

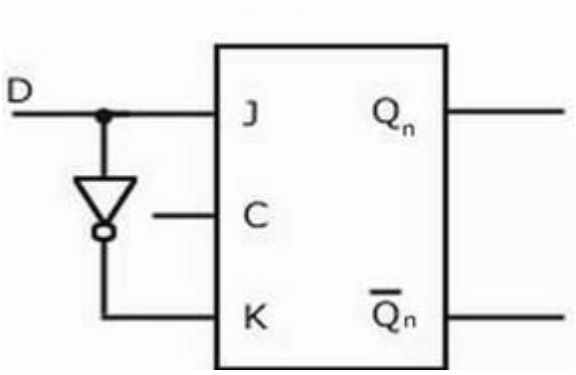


J	K	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	$\overline{Q}_0$ (toggles)

LOGICDIAGRAMOFRSFLIP FLOPS **TRUTHTABLE**

DFlipflop

It has only one input called as D input or Data input. The input data is transferred to the outputafteraclockisapplied.DFFcanbederivedfromJKFFbyusingJinputasDinput and J is inverted and fed to K input.

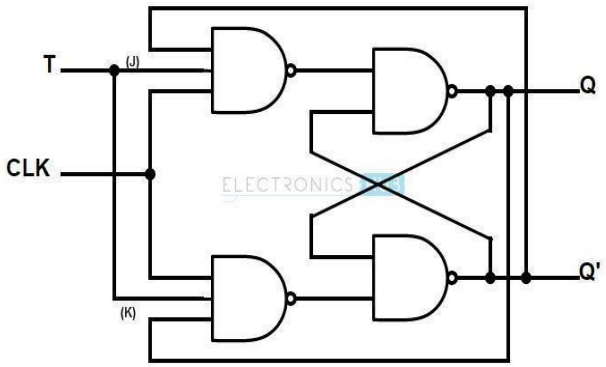


Q	D	Q(t+1)
0	0	0
0	1	1
1	0	0
1	1	1

LOGICDIAGRAM **TRUTHTABLE**

TFlipFlop

TstandsforToggle.Theoutputtoggleswhenaclockpulseisapplied.TFFcanbederived from JK FF by shorting J and K input.



T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

T FLIP FLOP LOGIC DIAGRAM **TRUTHTABLE**

PRE-EXPERIMENT QUESTIONS:-

1. Whatispropagationdelaytime?
2. HowisJKFFmadeto toggle?

PROCEDURE:-

- TestallcomponentsandICpackagesusingdigital ICtesterand multimeter
- SetupFFusingGatesandverifytheirtruthtables
- VerifytheTruthtablesof 7473,7474,and7476ICs

POST-EXPERIMENTQUESTIONS:-

1. HowmanyFFarein7475IC?
2. HowmanyFFarerequiredtoproduceadivide-by-128device?

LABEXPERIMENT6

OBJECTIVE: DesignAndImplementationOfShiftRegisters(SISO&SIPO)

EQUIPMENTS&COMPONENTSREQUIRED:

SL.No.	Equipments	Specification	Quantity
1	DigitalICTrainerkit	-	1
2	DigitalMultimeter	-	1

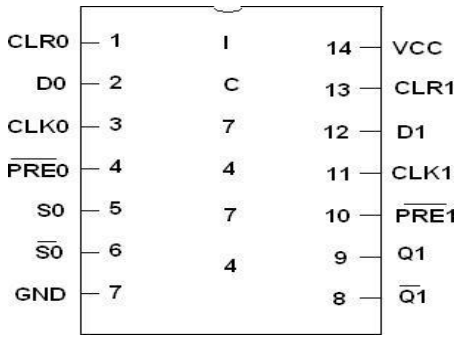
SL.No.	COMPONENT	SPECIFICATION	QTY.
1.	DFLIPFLOP	IC7474	2
2.	ICTRAINERKIT	-	1
3.	PATCHCORDS	-	15

BRIEFDESCRIPTION:

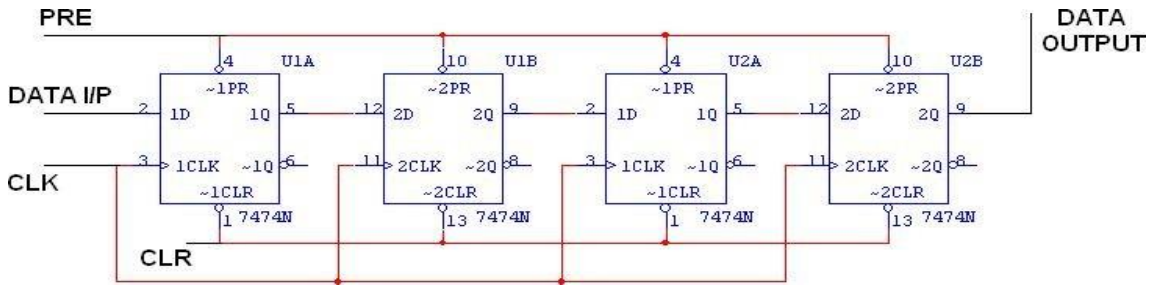
Aregisteriscapableofshiftingitsbinaryinformationinoneorbothdirectionsisknown as shift register. The logical configuration of shift register consist of a D-Flip flop cascaded with output of one flip flop connected to input of next flip flop. All flip flops receive common clock pulses which causes the shift in the output of the flip flop. The simplest possible shift register is one that uses only flip flop. The output of a given flip flopisconnected totheinputofnextflipflopoftheregister. Each clockpulseshiftsthe content of register one bit position to right.

(a) SerialinserialoutShiftRegister

(b) SerialinparalleloutShiftRegister



PINDIAGRAM

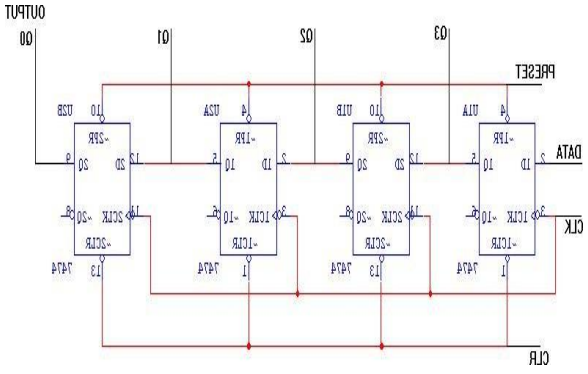


LOGICDIAGRAM:SERIALINSERIALOUT

CLK	Serialin	Serialout
1	1	0
2	0	0
3	0	0
4	1	1
5	X	0
6	X	0
7	X	1

TRUTHTABLE

SERIALINPARALLELOUT



CLK	DATA	OUTPUT			
		Q_A	Q_B	Q_C	Q_D
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	1
4	1	1	0	0	1

LOGICDIAGRAM **TRUTHTABLE**

PRE-EXPERIMENTQUESTIONS:-

1. StatethefeaturesofIC7495.
2. StatethefeaturesofIC74195.

PROCEDURE:-

- Connectionsaregivenaspercircuitdiagram.
- Logicalinputsaregivenaspercircuitdiagram.
- Observetheoutputandverifythetruthtable.

POST-EXPERIMENTQUESTIONS:-

1. Howcanweuseshiftregistersinserialcommunications?Explain.
2. ListtheICswhichareusedas8bitSISO,SIPO,PISO,PIPOmodesandasa bidirectional shift register.

LABEXPERIMENT:7

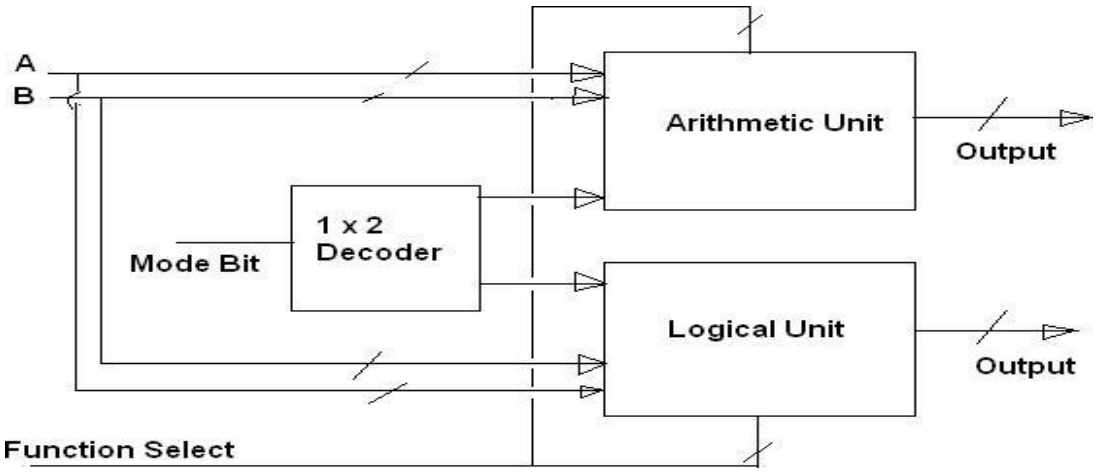
OBJECTIVE: Designandimplementationofan8bitarithmeticlogic unit.

EQUIPMENTS&COMPONENTSREQUIRED:

SL.No.	Equipments	Specification	Quantity
1	DigitalICTrainerkit	-	1
2	DigitalMultimeter		1

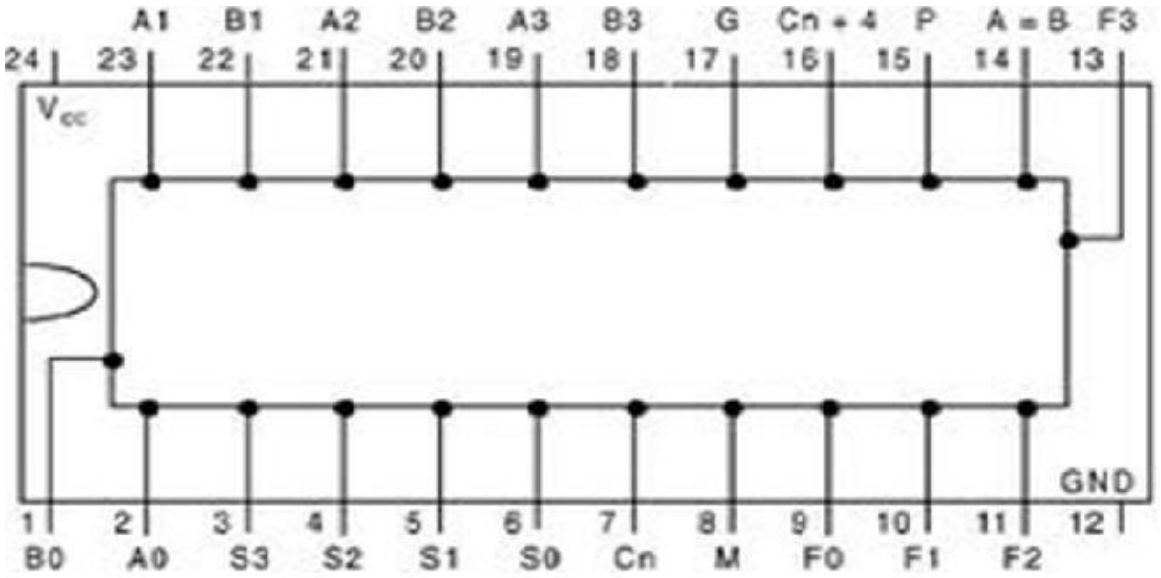
SL.No.	COMPONENT	SPECIFICATION	QTY.
1.	ALUIC	IC7474	2
2.	PATCHCORDS	-	15

BRIEFDESCRIPTION:
ALUstandsforthearithmeticandlogicalunitandisoneoftheimportantunitinalmostall the calculating machine these days be it with the hand-held mobile, or computers. All the computational work in the system are carried out by this unit.The typical ALU sizes are :
4-bit ALU : ALU that processes two 4-bit numbers.
8-bitALU:ALUthatprocessestwo8-bitnumbers.
StillinthelatestsystemsALUsizesare16,32,64-bitetc.Figure-1showstheblockdiagram of a typical ALU.



BlockDiagramofALU:

In figure-1,the1x2selectorontheleftisasamodeselectortoselectoneofthetwounitsi.e. either the arithmetic unit or the logical unit. The function select lines are then used to select one of the many functions of arithmetic or the logical type.
MSIpackageforALU:-IC74181a4-bitArithmeticandlogicalunit:



PINDIAGRAMOFIC74181ALU

SN54/74LS181

FUNCTION TABLE

MODE SELECT INPUTS				ACTIVE LOW INPUTS & OUTPUTS		ACTIVE HIGH INPUTS & OUTPUTS	
S_3	S_2	S_1	S_0	LOGIC (M = H)	ARITHMETIC** (M = L) (C_n = L)	LOGIC (M = H)	ARITHMETIC** (M = L) (C_n = H)
L	L	L	L	$\overline{A}$	A minus 1	$\overline{A}$	A
L	L	L	H	$\overline{AB}$	$\overline{AB}$ minus 1	$\overline{A}$ + B	A + $\overline{B}$
L	L	H	L	A + B	AB minus 1	AB	A + B
L	L	H	H	Logical 1 minus 1		Logical 0 minus 1	
L	H	L	L	$\overline{A}$ + B	A plus (A + $\overline{B}$)	$\overline{AB}$	A plus AB
L	H	L	H	$\overline{B}$	AB plus (A + B)	B	(A + B) plus AB
L	H	H	L	A ⊕ B	A minus B minus 1	A ⊕ B	A minus B minus 1
L	H	H	H	$\overline{A}$ + B	A + B	$\overline{AB}$	AB minus 1
H	L	L	L	AB	A plus (A + B)	$\overline{A + B}$	A plus AB
H	L	L	H	A ⊕ B	A plus B	A ⊕ B	A plus $\overline{B}$
H	L	H	L	B	AB plus (A + B)	B	(A + B) plus AB
H	L	H	H	A + B	A + B	AB	AB minus 1
H	H	L	L	Logical 0 A plus A*		Logical 1 A plus A*	
H	H	L	H	AB	A$\overline{B}$ plus A	A + B	(A + $\overline{B}$) plus A
H	H	H	L	AB	AB plus A	A + B	(A + B) Plus A
H	H	H	H	A	A	A	A minus 1

L = LOW Voltage Level
H = HIGH Voltage Level
*Each bit is shifted to the next more significant position
**Arithmetic operations expressed in 2s complement notation

PRE-EXPEIMENTQUESTIONS:-

1. WhatisDigitizing?

PROCEDURE:-

- KeepthedatasheetofIC74181ready.
- InserttheIContheBreadboard.
- Makeconnectionsasshowninfigure.
- Verifytheconnections

POST-EXPEIMENTQUESTIONS:-

1.WhichBooleanoperatorcombinessearchtermssothateachsearchresultcontainsall term ?

LABEXPERIMENT8

OBJECTIVE: Design a data path of a computer from its register transfer language description.

EQUIPMENTS&COMPONENTSREQUIRED:

SL.No.	Equipment's	Specification	Quantity
1	DigitalICTrainerkit	-	1
2	DigitalMultimeter		1

SL.No.	Components	Specification	Quantity
1	DigitalICs	7400,7402,7404, 7408,7432,7486.	1each
2	Patchcords	-	6

BRIEFDESCRIPTION:

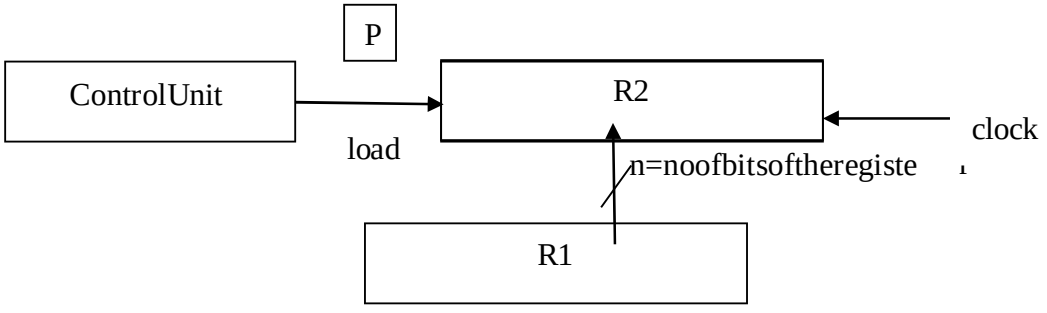
Thesymbolicnotationusedtodescribethemicro-operationtransfers amongregistersiscalled Register Transfer Language. The term "register transfer" implies the availability of hardware logic circuitsthatcan perform astated micro-operationandtransfer theresultoftheoperation to the same or another register.

Astatementthatspecifiesaregistertransferimpliesthatcircuitsareavailablefromtheoutputs of the source register to the inputs of the destination register and that the destination register has a parallel load capacity. If the transfer is to occur under a predetermined condition i.e.

If(P=1)thenR2←R1

WherePisthecontrolsignal generatedinthecontrolsection.AControlFunctionisaBoolean variable that is equal to 0 or 1

P:R2←R1



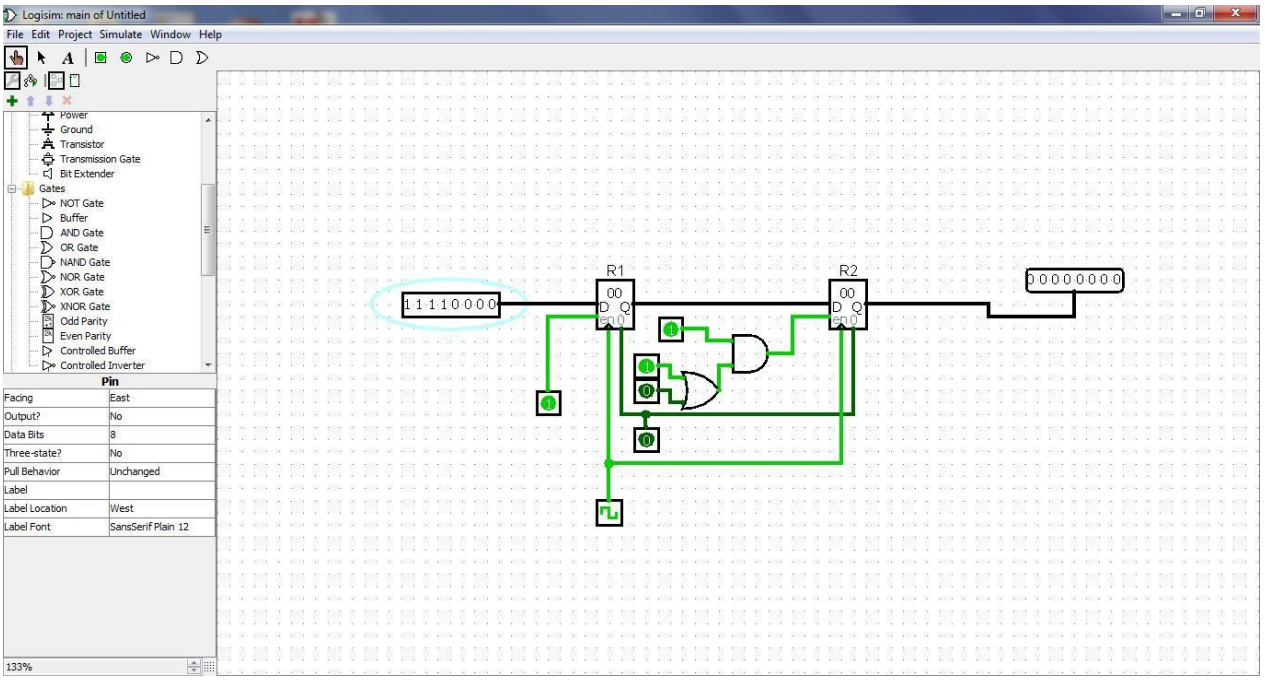
BLOCKDIAGRAM

PRE-EXPERIMENTQUESTIONS:-

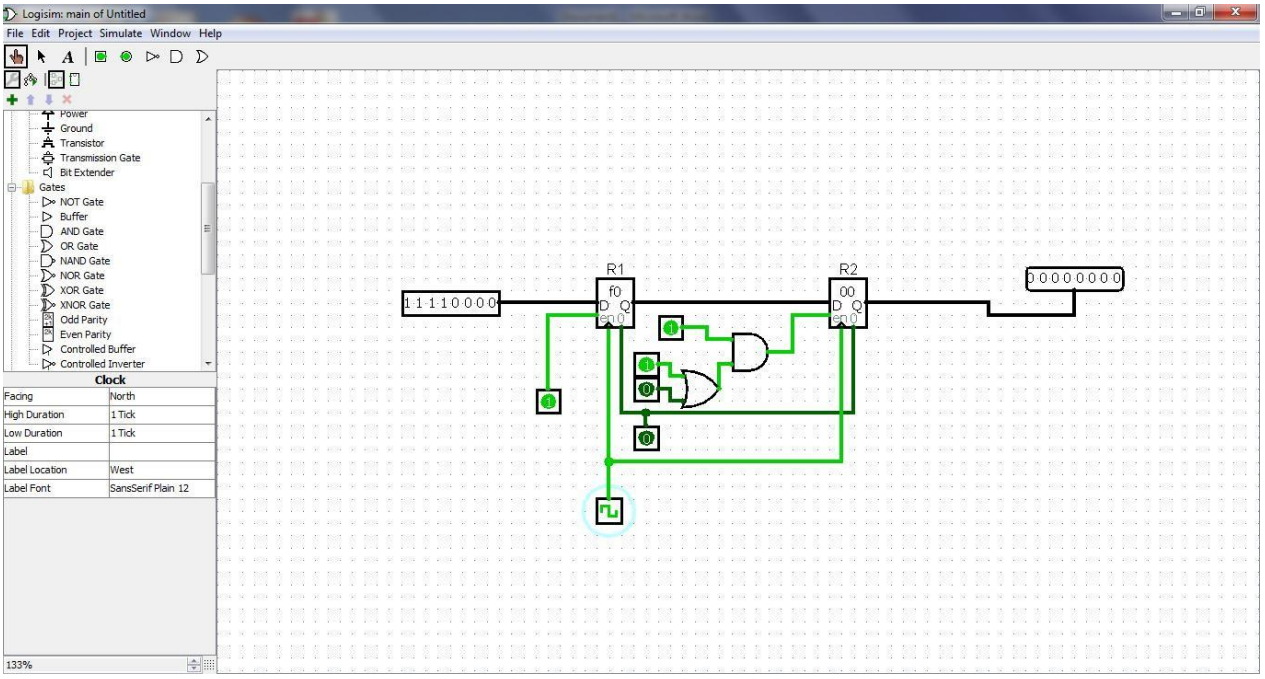
1. Whatisthesignificance ofDataPath?
2. WhatisRegisterTransfer Logic?

PROCEDURE:-

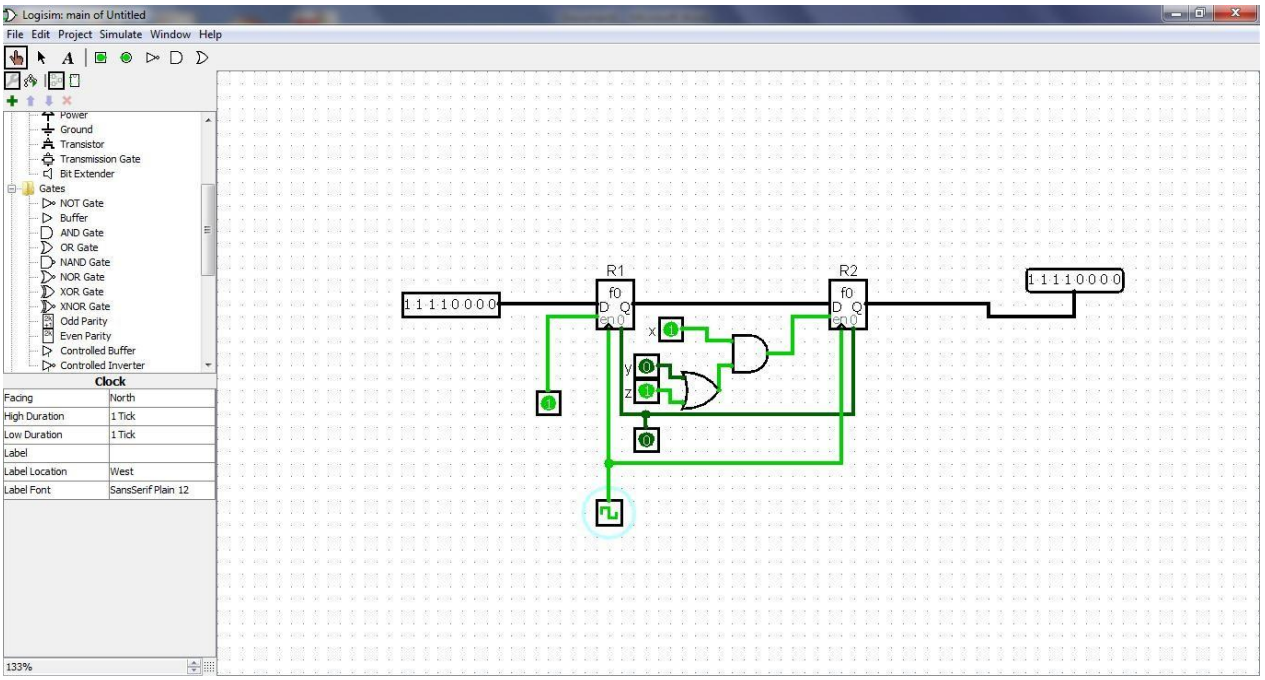
DatabitsReadytobetransferred



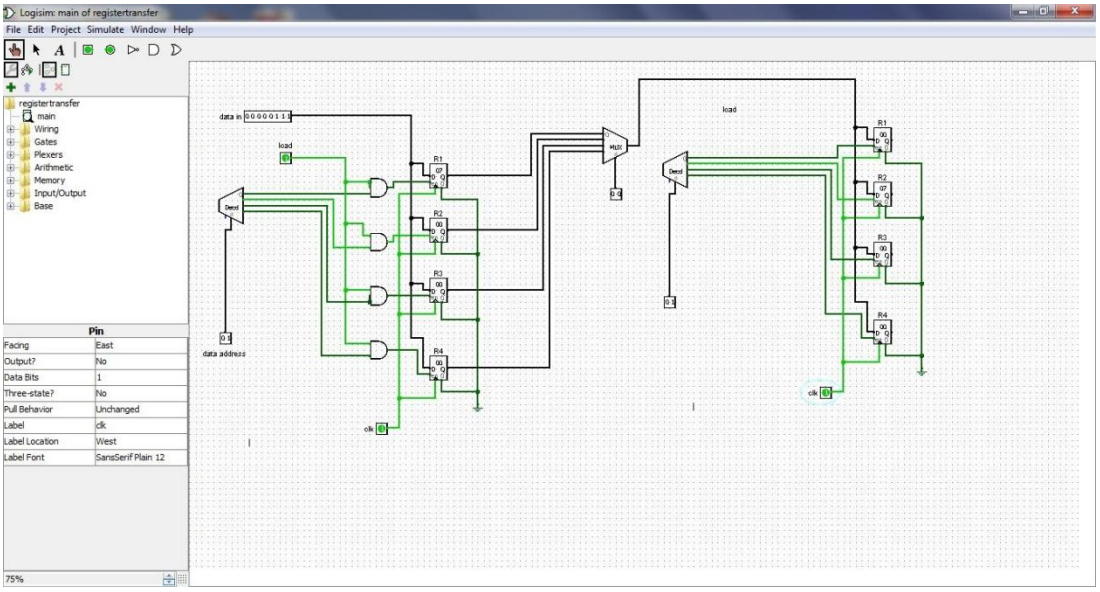
InthefirstclocktickRegisterR1storesthevalue



Inthenextclockpulsethevaluegetstransferredtoregister R2 if its Enable input is high, i.e. its control function x.(y+z) =1



RegisterTransferusingBUS



POST-EXPERIMENTQUESTIONS:-

1. Howadatapathisdesigned?
2. Designadatapath for$R_3 \leftarrow R_2+R_1$

LABEXPERIMENT9

OBJECTIVE:Designadatapathofacomputerfromitsregistertransferlanguagedescription.

EQUIPMENTS&COMPONENTSREQUIRED:

Registers
ANDGates
OR Gate
ConnectingWires

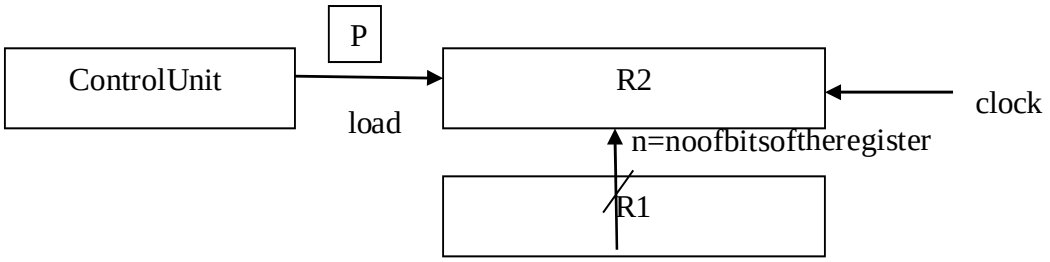
BREIFDESCRIPTION:

Thesymbolicnotationusedtodescribethemicro-operationtransfersamongregistersis called Register Transfer Language.
Theterm“registertransfer”impliestheavailabilityofhardwarelogiccircuitsthatcanperform astatedmicro-operationandtransfertheresultoftheoperationtothesameoranotherregister.
Astatementthatspecifiesaregistertransferimpliesthatcircuitsareavailablefromtheoutputs of the source register to the inputs of the destination register and that the destination register has a parallel load capacity.
Ifthetransferistooccurunderapredeterminedconditioni.e

If(P=1)thenR2←R1

WherePisthecontrolsignal generatedinthecontrolsection.AControlFunctionisaBoolean variable that is equal to 0 or 1

P:R2←R1



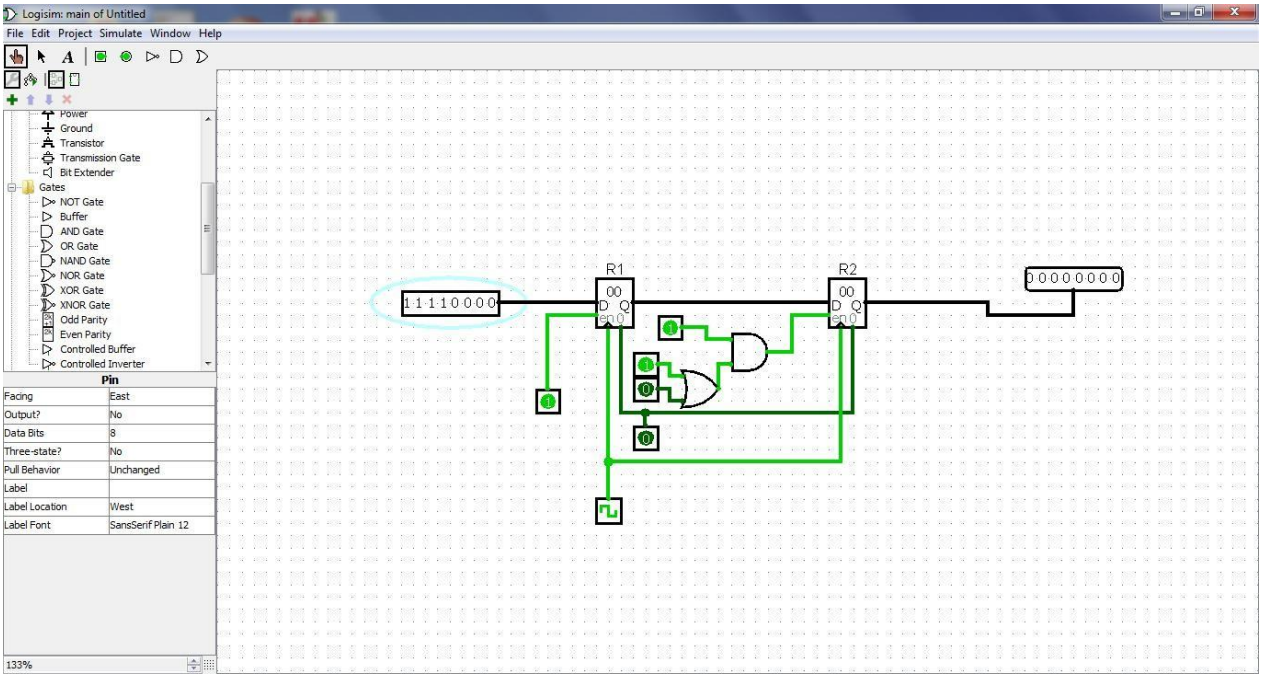
BLOCKDIAGRAM

PRE-EXPERIMENTQUESTIONS:-

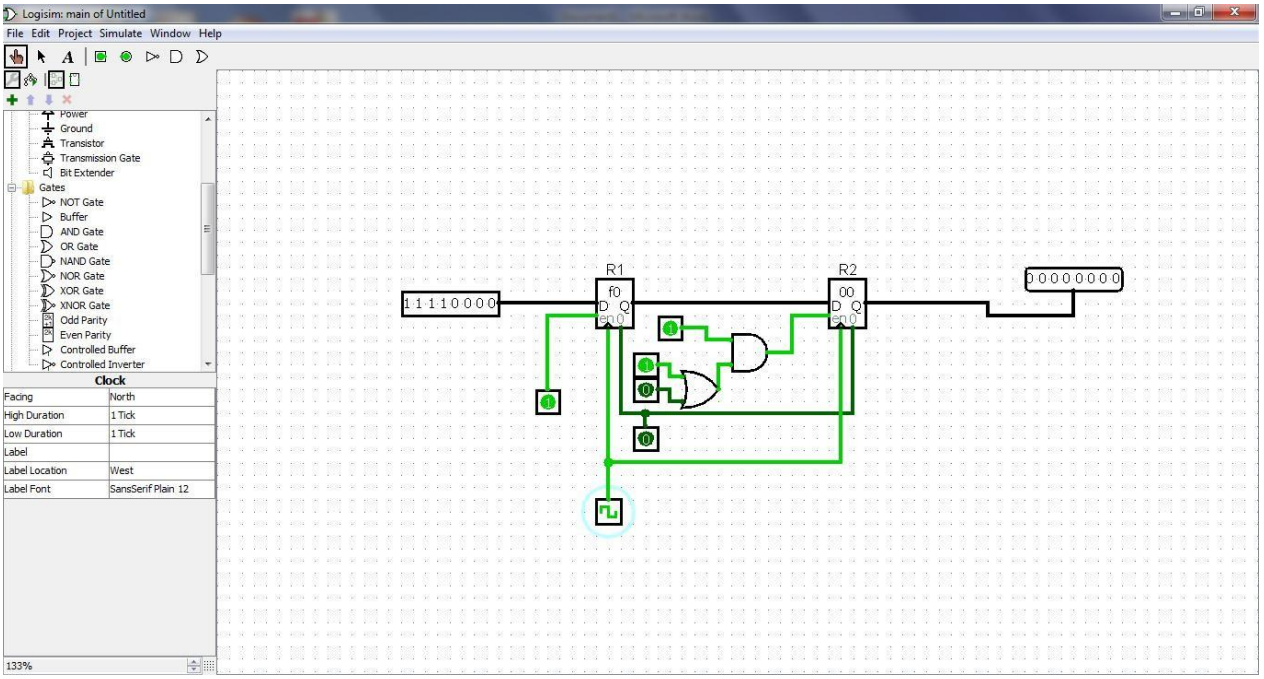
1. WhatisRTL?
2. Whatistheminimumno.ofregistersneededintheinstructionsetarchitectureofthe processor to compile a code with 3 operands?

PROCEDURE:-

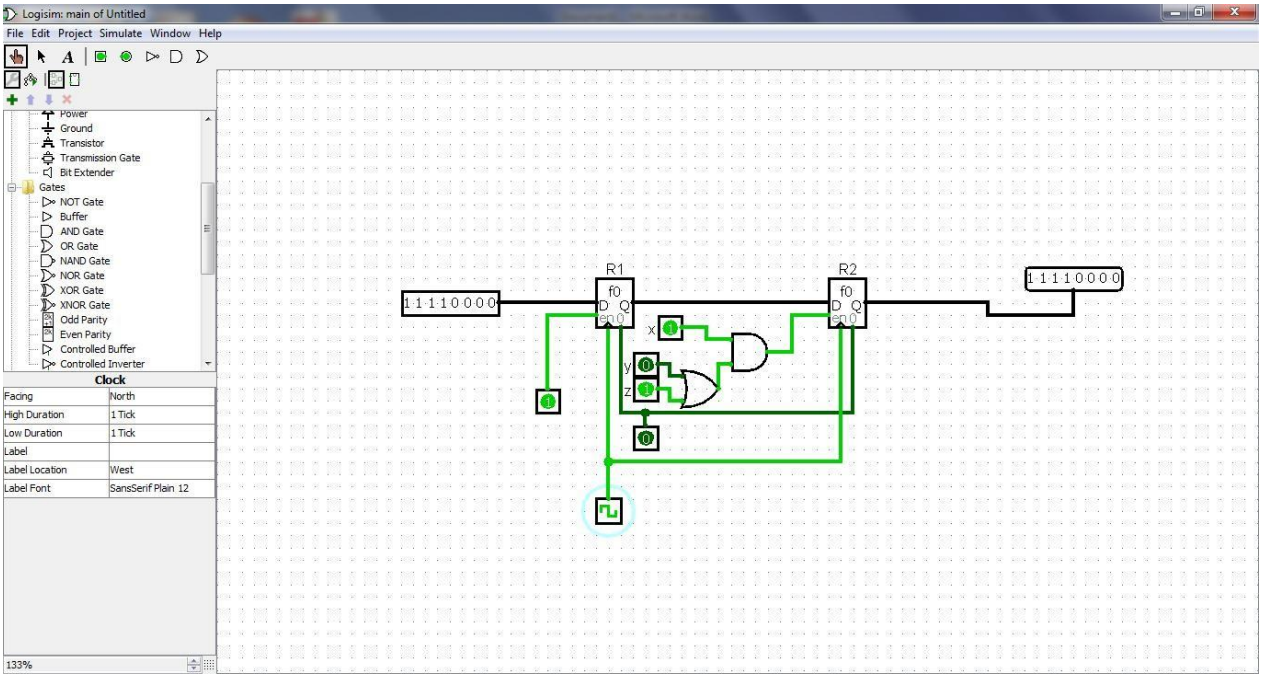
DatabitsReadytobetransfered

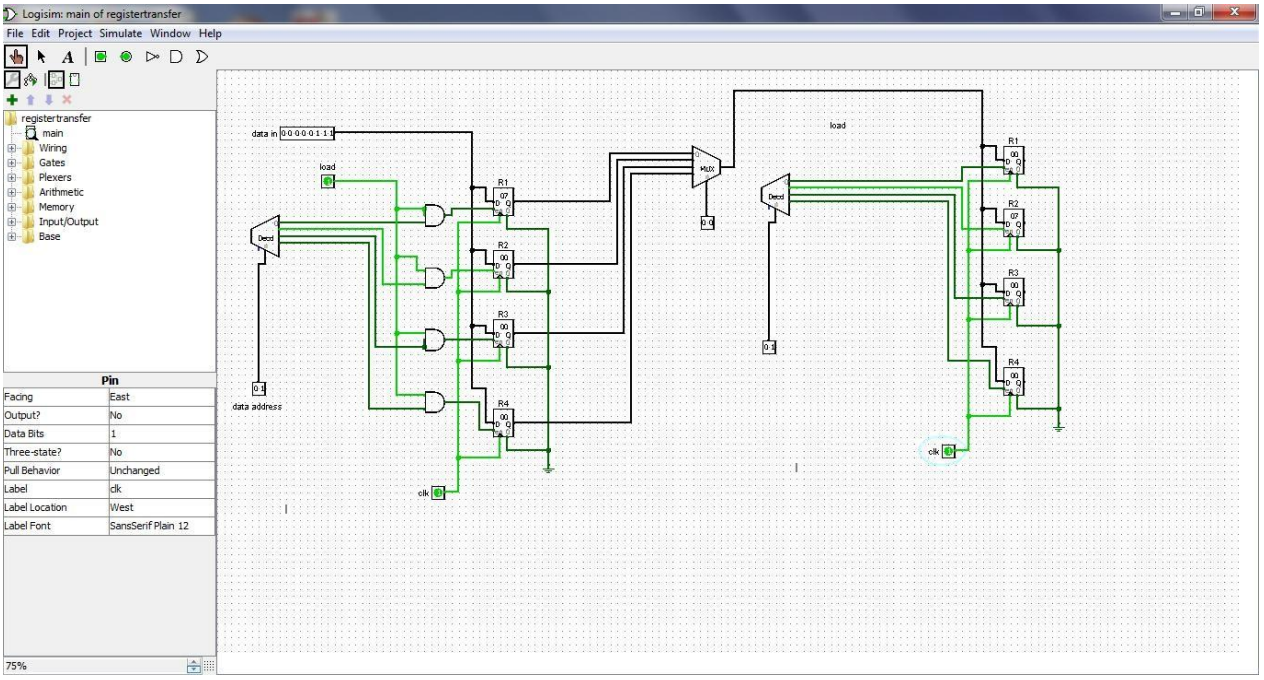


InthefirstclocktickRegisterR1storesthevalue



InthenextclockpulsethevaluegetstransferredtoregisterR2ifitsEnableinputishigh,i.e its control function x.(y+z) =1





REGISTERTRANSFERUSINGBUS

POST-EXPERIMENTQUESTION:-

1. WhatistheamountofROMneededtoimplementa4bitmultiplier.

LABEXPERIMENT10

OBJECTIVE: Design the control unit of a computer using hardwired based on its registertransfer language description.

COMPONENTSREQUIRED:
1.Registers
2. ANDGates
3. ORGate
4. Memory
5. DecodersandMultiplexers
6. ConnectingWires,etc.

BREIFDESCRIPTION:

Thecontrolunit(CU)isacomponentofacomputer'scentralprocessingunit(CPU)thatdirects the operation of the processor. It tells the computer's memory, arithmetic/logic unit and input and output devices on how to respond to a program's instructions.

It directs the operation of the other units by providing timing and control signals. Most computer resources are managed by the CU. It directs the flow of data between the CPU and theotherdevices.TheControlUnit(CU)isdigitalcircuitrycontainedwithintheprocessorthat coordinatesthesequence ofdatamovementsinto,outof,andbetweenaprocessor'smanysub- units. The result of these routed data movements through various digital circuits (sub-units) withintheprocessorproducesthemanipulateddataexpectedbyasoftware instruction(loaded earlier, likely from memory). It controls (conducts) data flow inside the processor and additionallyprovidesseveralexternalcontrolsignalstotherestofthecomputertofurtherdirect data and instructions to/from processor external destination's (i.e. memory).

PRE-EXPERIMENTQUESTION:-

1.Whyisregisterrenamingdoneinpipelined processor.

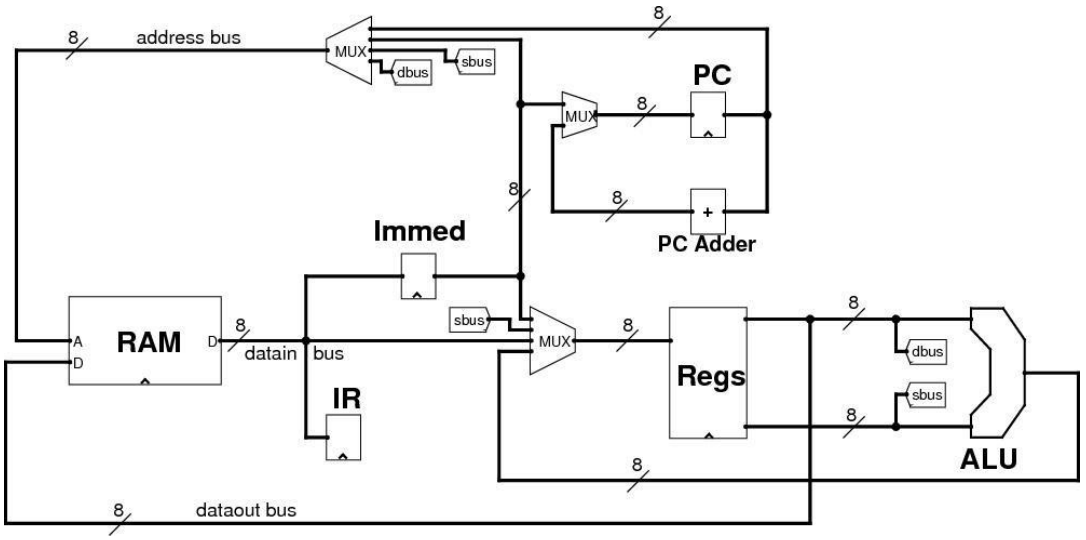
PROCEDURE:-

TheCPUhasan8-bitdatabusandan8-bitaddressbus,soitcanonlysupport256bytesof memory to hold both instructions and data.

- Internally,therearefour8-bitregisters,R0toR3,plusanInstructionRegister,the Program Counter, and an 8-bit register which holds immediate values.

- TheALU isthesameonethatwedesigned lastweek. It performsthefouroperations AND,OR,ADDandSUBontwo8-bitvalues,andsupportssignedADDsandSUBs.
- TheCPUisaload/storearchitecture:datahastobebroughtintoregistersfor manipulation, as the ALU only reads from and writes back to the registers.
- TheALUoperationshavetwooperands:oneregisterisasourceregister,andthe second register is both source and destination register, i.e. destination register = destination register OP source register.

- Allthejumpoperationsperformabsolutejumps;therearenoPC-relativebranches. There are conditional jumps based on the zeroness or negativity of the destination register, as well as a "jump always" instruction.
- ThefollowingdiagramshowsthedatapathsintheCPU:



- The*dbus*and*sbus*labelsindicatethelinescomingoutfromtheregisterfilewhich hold the value of the destination and source registers.
- NotethedataloopinvolvingtheregistersandtheALU,whoseoutputcan onlygo back into a register.
- Thedataoutbusisonlyconnectedtothe*dbus*line,sotheonlyvaluewhich canbe written to memory is the destination register.
- Alsonotethattherearеonly3multiplexors:
  - the address bus multiplexor can get a memory address from the PC, the immediateregister(fordirectaddressing),orfromthesourceordestination registers (for register indirect addressing).
  - thePCmultiplexoreitherletsthePCincrement,orjumptothevalueinthe immediate register.
  - the multiplexor in front of the registers determines where a register write comesfrom:theALU,theimmediateregister,anotherregisterorthedatabus.

Aswehavedecidedthehardwaretobeusedinourdesignnowwehavetodecidethe instruction set for our computer.

InstructionSet

- Halfoftheinstructionsintheinstructionset fitintoonebyte:

op1	op2	Rd	Rs
2	2	2	2

- Theseinstructionsareidentifiedbya0inthemost-significantbitintheinstruction, i.e.*op1*=0X.
- The4bitsofopcode aresplitinto *op1*and *op2*
- *Rd*isthedestinationregister,and*R*sisthesource register.
- Theotherhalfoftheinstructionsetaretwo-byteinstructions.Thefirstbytehasthe same format as above, and it is followed byan 8-bit constant or immediate value:

op1	op2	Rd	Rs	immediate
2	2	2	2	8

- Thesetwo-byteinstructionsareidentifiedbya1inthemost-significantbitinthe instruction, i.e. *op1* = 1X.
- With4operationbits,thereare16instructions:

op1	**op2**	**Mnemonic**	**Purpose**
00	00	ANDRd,Rs	Rd=RdANDRs
00	01	ORRd,Rs	Rd=RdORRs
00	10	ADDRd,Rs	Rd=Rd+Rs
00	11	SUBRd, Rs	Rd=Rd –Rs
01	00	LWRd, (Rs)	Rd=Mem[Rs]
01	01	SWRd,(Rs)	Mem[Rs]=Rd
01	10	MOVRd,Rs	Rd=Rs
01	11	NOP	Donothing
10	00	JEQRd,immed	PC=immed ifRd ==0
10	01	JNERd,immed	PC=immed ifRd !=0
10	10	JGTRd,immed	PC=immed ifRd >0
10	11	JLTRd,immed	PC=immed ifRd <0
11	00	LWRd,immed	Rd=Mem[immed]
11	01	SWRd,immed	Mem[immed]=Rd
11	10	LIRd, immed	Rd=immed
11	11	JMPimmed	PC=immed

- NotetheregularityoftheALUoperationsandthejumpoperations:wecanfeed the *op2* bits directly into the ALU, and use *op2* to control the branch decision.
- Therestoftheinstructionsetislessregular,whichwillrequirespecialdecodingfor certain of the 16 instructions.

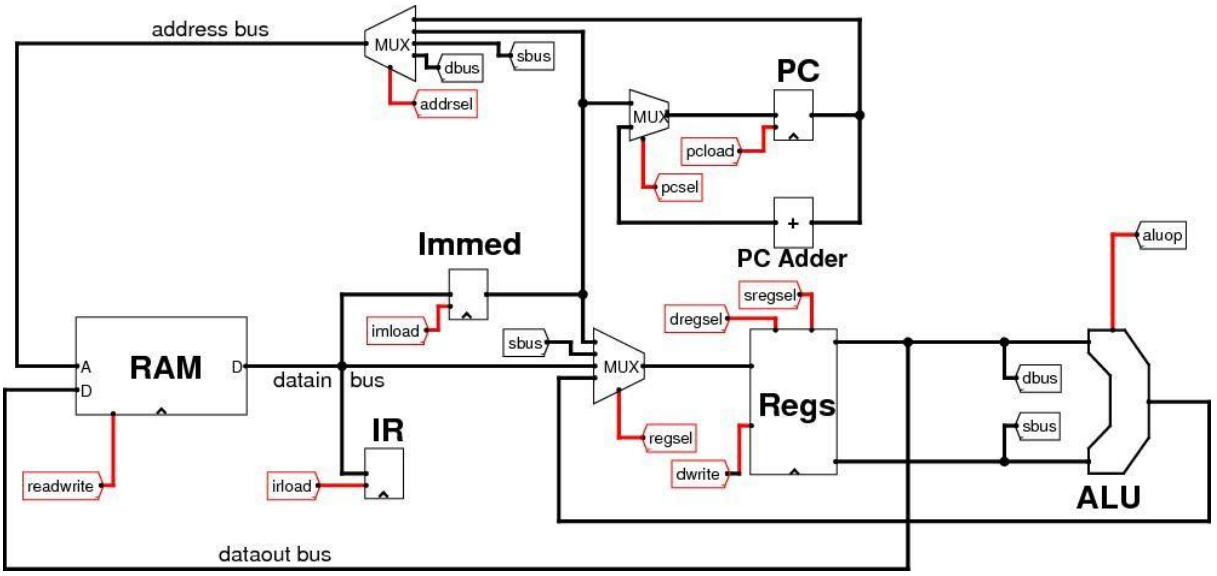
InstructionPhases

- TheCPUinternallyhasthreephasesfortheexecutionofeachinstruction.

- Onphase0,theinstructionisfetchedfrommemoryandstoredintheInstruction Register.
- On phase 1, if the fetched instruction is a two-byte instruction, the second byte is fetchedfrommemoryandstoredintheImmediateRegister.Forone-byteinstructions, nothing occurs in phase 1.
- Onphase2,everythingelseisdoneasrequired,whichcaninclude:
  - anALUoperation,readingfromtworegisters.
  - ajumpdecisionwhichupdatesthePC.
  - aregisterwrite.
  - aread fromamemorylocation.
  - awritetoamemorylocation.
- Afterphase2,theCPUstartsthenextinstructioninphase0.

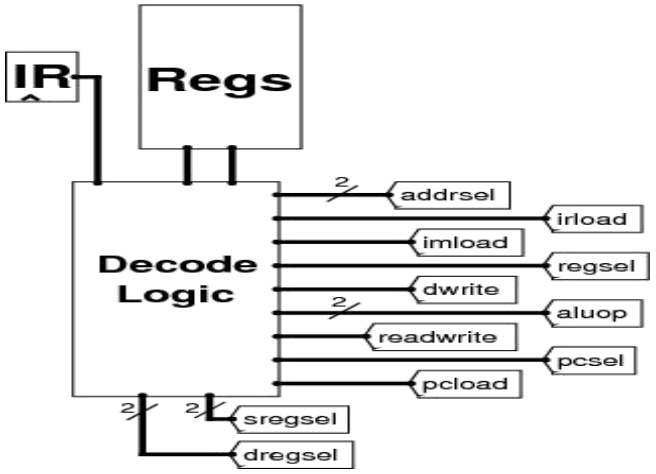
CPUControlLines

- BelowisthemainCPUdiagramagain,thistimewiththecontrollinesshown.



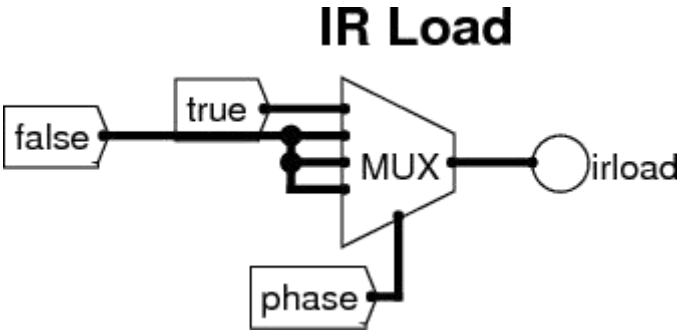
- Thereareseveral1-bitcontrollines:
  - *pcsel*,incrementPCorloadthejumpvaluefromtheImmediateRegister.
  - *pcload*,loadthePCwithanewvalue,ordon't loadanewvalue.
  - *irload*,loadtheInstructionRegisterwithanewinstruction.
  - *imload*,loadtheImmediateRegisterwithanewvalue.
  - *readwrite*,readfrommemory,orwritetomemory.
  - *dwrite*,writeavaluebacktoaregister,ordon'twritea value.
- Therearealsoseveral2-bitcontrollines:
  - *addrsel*,selectanaddressfromthePC,theImmediateRegister,thesource register or the destination register.
  - *regsel*,selectavaluetowritetoaregisterfromtheImmediateRegister, another register, the data bus or from the ALU.
  - *dregsel*and*sregsel*,selecttworegisterswhosevaluesaresenttotheALU.
  - *aluop*,whicharethe*op2*bitsthatcontroltheoperationoftheALU.

- The values for all of these control lines are generated by the Decode Logic, which getsasinputthevaluefromtheInstructionRegister,andthezero&negativelinesof the destination register.



PhaseZero

- Onphasezero,thePC'svaluehastobeplacedontheaddressbus,sothe*addrsel*line mustbe 0.The*irload*lineneedstobe1sothatthe IR isloadedfrom the*datain*bus. Finally, thePCmust be incremented incase we need to fetch animmediatevalue in phase 1.
- Allofthiscanbedoneusingmultiplexorswhichoutputdifferentvaluesdependingon the current phase. Here is the control logic for the *irload* line.



- Weonlyneed toloadtheIRonphase0, sowecanwiretruetothe0inputof the*irload*multiplexor,andfalsetotheotherinputs.**Note:**input11(i.e.decimal3)to the multiplexor is never used, as we never get to phase 3, but Logisim wants all multiplexor inputs to be valid.
- Anotherwaytolookateachphaseisthevaluewhichneedstobesetforeachcontrol line, for each instruction.
- Forphasezero,thesecontrollinevaluescanbesetforallinstructions:

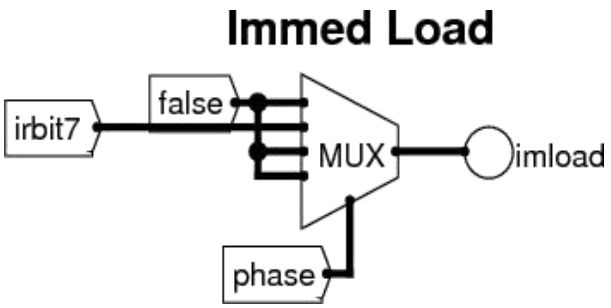
op 1	op 2	instruct	pcsel	pcload	Irload	imload	Rw	dwrite	jumpsel	addrsel	regsel	dreg	sreg	aluop

xx	xx	all	1	1	1	0	0	0	0	0	x	x	x	x

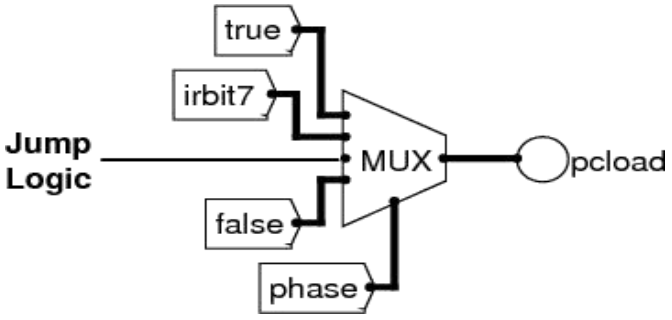
- 'x'standsfor anyvalue,i.e.accept anyopcodevalue,outputanycontrolline value.

PhaseOne

- On phase 1, we need to load the Immediate Register with a value from memory if the*irbit7*fromtheIRistrue.ThePC'svaluehastobeplacedontheaddressbus,so the *addrsel* line must be 0. The *imload* line needs to be 1 so that the Immediate Registerisloadedfromthe*datain*bus.Finally, thePCmustbeincrementedsothat we are ready to fetch the next instruction on the next phase 0.



- The*imload*logicisshownabove.Itisverysimilartothe*irload*logic,butthistimean enable value is output only on phase 1, and only if the *irbit7* is set.



- Someofthe*pcload*logicisshownabove.ThePCisalwaysincrementedatphase0.It is incremented at phase 1 if *irbit7* is set, i.e. a two-byte instruction. Finally, the PC can be loaded with an immediate value in phase 2 if we are performing a jump instruction and the jump test is true. We will come back to the jump logic later.
- Wecantabulatethevaluesofthecontrollinesforphase1.Thistime,whatisoutput depends on the top bit of the *op1* value:
-

op 1	**op 2**	**instruct**	**pcsel**	**pcload**	**Irload**	**imload**	**Rw**	**dwrite**	**jumpsel**	**addrsel**	**regsel**	**dreg**	**sreg**	**aluop**

0xxx	all	x	0	0	0	0	0	0	0	x	x	x	x
1xxx	all	1	1	0	1	0	0	0	0	x	x	x	x

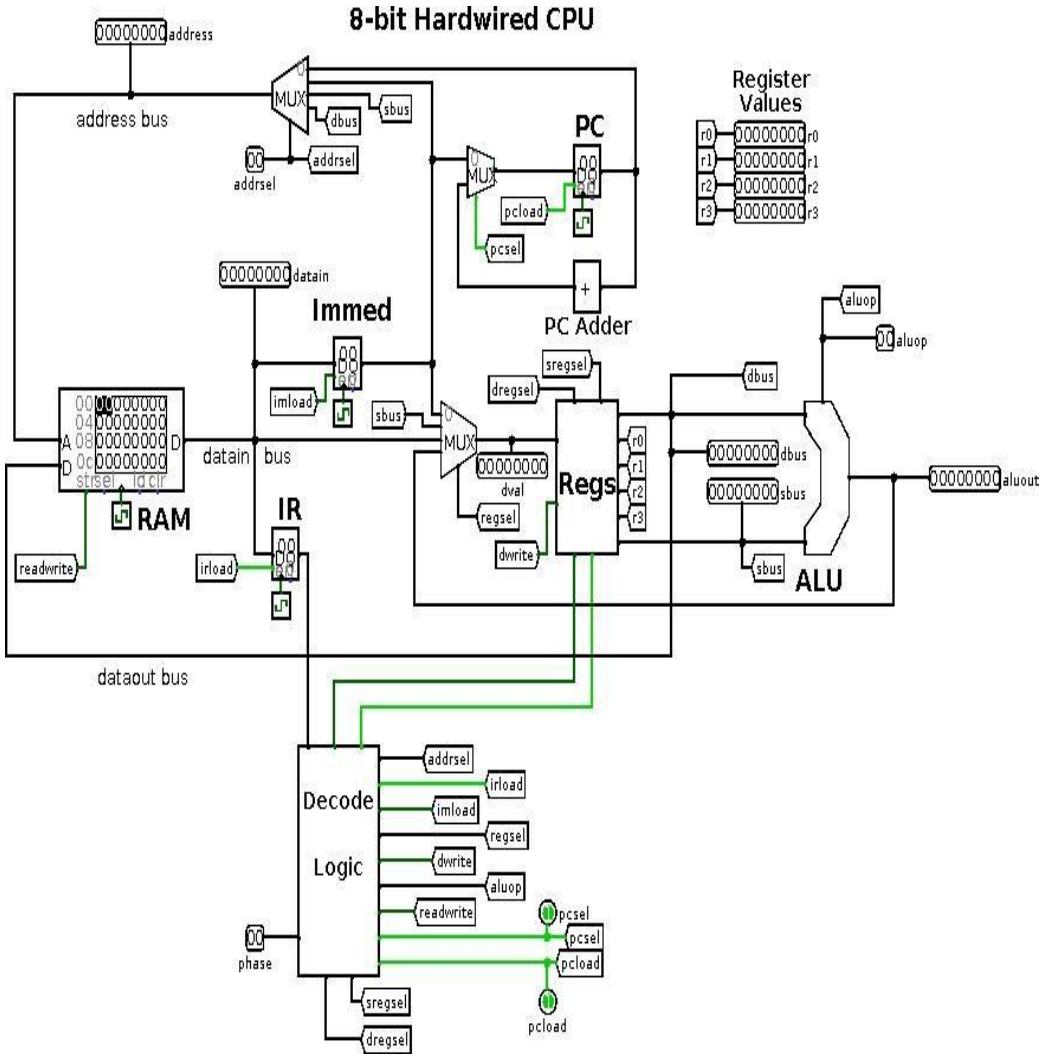
PhaseTwo

- Here,thevalues ofthecontrollinesdepend heavilyonwhat specific instructionwe areperforming.Here'sthetableofcontrollineoutputsdependingontheinstruction:

op1	op2	instruct	pcsel	pcload	irload	imload	rw	dwrite	addrsel	regsel	dreg	sreg	aluop
00	00	ANDRd,Rs	x	0	0	0	0	1	x	3	Rd	Rs	op2
00	01	ORRd, Rs	x	0	0	0	0	1	x	3	Rd	Rs	op2
00	10	ADDRd,Rs	x	0	0	0	0	1	x	3	Rd	Rs	op2
00	11	SUBRd,Rs	x	0	0	0	0	1	x	3	Rd	Rs	op2
01	00	LWRd,(Rs)	x	0	0	0	0	1	2	2	Rd	Rs	x
01	01	SWRd,(Rs)	x	0	0	0	1	0	3	x	Rd	Rs	x
01	10	MOVRd, Rs	x	0	0	0	0	1	x	1	Rd	Rs	x
01	11	NOP	x	0	0	0	0	0	x	x	x	x	x
10	00	JEQRd,immed	0	J	0	0	0	0	x	x	Rd	x	op2
10	01	JNERd,immed	0	J	0	0	0	0	x	x	Rd	x	op2
10	10	JGTRd,immed	0	J	0	0	0	0	x	x	Rd	x	op2
10	11	JLTRd,immed	0	J	0	0	0	0	x	x	Rd	x	op2
11	00	LWRd,immed	x	0	0	0	0	1	1	2	Rd	x	x
11	01	SWRd,immed	x	0	0	0	1	0	1	x	Rd	x	x
11	10	LIRd,immed	x	0	0	0	0	1	x	0	Rd	x	x
11	11	JMPimmed	0	1	0	0	0	0	x	x	x	x	x

- Tomakethecontrollinelogicassimpleaspossible,aCPUdesignerisalways striving for regularity. However, this is often in conflict with the desired CPU functionality.
- Fromthetableabove,theALUinstructions(*op1*=00)andthejumpinstructions (*op1*=10) are nice and regular. All the *op1*=1x instructions use the Immediate Register, while the *op1*=0x instructions don't.
- Wecanalwaystie*dregsel*to*Rd*fromtheinstruction,andthesame goes for*sregsel=Rs*and*aluop=op2*.And*irload*and*imload*arealways0forphase2.
- Withtheremainingcontrollines,theregularitiescease.

Puttingthisallbacktogether,wenowhavethisdevice:



POST-EXPERIMENTQUESTION:-

1.WhichDMAtransfermodeandinterrupthandlingmechanismwillenablethe highest I/O bandwidth.