BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
K. K. BIRLA Goa Campus
Second Semester 2012-2013
IS F242 Computer Organization

INSTRUCTIONS

1. The project weightage is 6% (18 marks) and the deadline is 23rd April 2013.
2. This is a team project and the list of team members is available in course page.
3. You are not allowed to take help in terms of suggestions and code from other teams.
4. You are not allowed to use somebody else's code which includes code from internet.
5. If found copied, all members of the team will get -18 marks (will subtract 18 marks from their lab marks). Please see section 4.2 and 4.3 of your course handout for further details.
6. This project will be evaluated according to the following criteria
   - Completeness
   - Correctness
   - Efficiency
   - Programming style
   - Demonstration and
   - Viva

STEPS TO FOLLOW FOR GETTING A PROJECT ASSIGNED

1. Each team should decide the preference order (you are suppose to give all the 12 preferences).
2. One person from the team should consolidate the preferences and send the same to my e-mail (biju@goa.bits-pilani.ac.in) at the earliest. [make sure you are sending your preferences to me before Monday 5 P.M]. [Please make sure only one person from a team is sending e-mail to me].
3. Allocation of project is on First Come First Serve basis. No default project allocation is available – i.e. if you are not sending e-mail before Monday 5 P.M, you are not going to do the project.

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
K. K. BIRLA Goa Campus, Second Semester 2012-2013
IS F242 Computer Organization, Project, Deadline: 23rd April 2013

Question #1

Design an optimized datapath for the architecture with following instruction formats and instructions. Implement the same as a **Single cycle implementation**.

R – type

4 bits	3 bits	3 bits	3 bits	3 bits
Opcode	Shftamt	Rd	Rt	Rs

I – type

4 bits	6 bits	3 bits	3 bits
Opcode	Immediate	Rt/Rd	Rs

J – type

4 bits	12 bits
Opcode	Immediate

Instructions:

ADD Example: Reg[R1] ← Reg[R2] + Reg[R3] ➔0001 XXX 001 011 010

SUB Example: Reg[R1] ← Reg[R2] - Reg[R3] ➔0010 XXX 001 011 010

DIVU Example: Hi, Lo ← Reg[R2] / Reg[R3] ➔0011 XXX XXX 011 010

MFHI Example: Reg[R1] ← Hi ➔0100 XXX 001 XXX XXX

MFLO Example: Reg[R1] ← Lo ➔0101 XXX 001 XXX XXX

MTLH (Move to Hi and Lo)

Example: Hi ← Reg[R2], Lo← Reg[R3] ➔0110 XXX XXX 011 010

SLL Example: Reg[R1] ← Reg[R2] << Shftamt (101) ➔0111 101 001 XXX 010

SRL Example: Reg[R1] ← Reg[R2] >> Shftamt (101) ➔1000 101 001 XXX 010

SRA Example: Reg[R1] ← Reg[R2] >> Shftamt (101) ➔1001 101 001 XXX 010

LW Example: Reg[R1] ← Memory[Reg[R2] + Sext(6 bits)] ➔1010 000011 001 010

SW Example: Memory[Reg[R2] + Sext(6 bits)] ← Reg[R1] ➔1011 000011 001 010

BNE Example:NZ flag=1 if Reg[R2] ≠Reg[R1],else NZ flag=0➔1100 000111 001 010

PC ← (PC+2) + Sext(6bits<<1)

LUI Example: Reg[R3] ← 8 bit immediate:8'd0 ➔1101 001100 011 X11

8 bit immediate value is IR[6:11]:IR[1:0]

ORI Example: Reg[R1] ← Reg[R2] OR Zext(6 bits) ➔1110 000011 001 010

J Example: PC← 3 bits MSB [(PC+2)]: Sext(12 bits<<1) ➔1111 000000000011

JALR Example: PC← Reg[R2], Reg[R3] ← PC+2 ➔0000 XXX 011 XXX 010

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
K. K. BIRLA Goa Campus, Second Semester 2012-2013
IS F242 Computer Organization, Project, Deadline: 23rd April 2013

Question #2

Design an optimized datapath for the architecture with following instruction formats and instructions. Implement the same as a **Multi cycle implementation**.

R – type

4 bits	3 bits	3 bits	3 bits	3 bits
Opcode	Shftamt	Rd	Rt	Rs

I – type

4 bits	6 bits	3 bits	3 bits
Opcode	Immediate	Rt/Rd	Rs

J – type

4 bits	12 bits
Opcode	Immediate

Instructions:

ADD Example: Reg[R1] ← Reg[R2] + Reg[R3] ➔0001 XXX 001 011 010

SUB Example: Reg[R1] ← Reg[R2] - Reg[R3] ➔0010 XXX 001 011 010

DIVU Example: Hi, Lo ← Reg[R2] / Reg[R3] ➔0011 XXX XXX 011 010

MFHI Example: Reg[R1] ← Hi ➔0100 XXX 001 XXX XXX

MFLO Example: Reg[R1] ← Lo ➔0101 XXX 001 XXX XXX

MTLH (Move to Hi and Lo)

Example: Hi ← Reg[R2], Lo← Reg[R3] ➔0110 XXX XXX 011 010

SLL Example: Reg[R1] ← Reg[R2] << Shftamt (101) ➔0111 101 001 XXX 010

SRL Example: Reg[R1] ← Reg[R2] >> Shftamt (101) ➔1000 101 001 XXX 010

SRA Example: Reg[R1] ← Reg[R2] >> Shftamt (101) ➔1001 101 001 XXX 010

LW Example: Reg[R1] ← Memory[Reg[R2] + Sext(6 bits)] ➔1010 000011 001 010

SW Example: Memory[Reg[R2] + Sext(6 bits)] ← Reg[R1] ➔1011 000011 001 010

BNE Example:NZ flag=1 if Reg[R2] ≠Reg[R1],else NZ flag=0➔1100 000111 001 010

PC ← (PC+2) + Sext(6bits<<1)

LUI Example: Reg[R3] ← 8 bit immediate:8'd0 ➔1101 001100 011 X11

8 bit immediate value is IR[6:11]:IR[1:0]

ORI Example: Reg[R1] ← Reg[R2] OR Zext(6 bits) ➔1110 000011 001 010

J Example: PC← 3 bits MSB [(PC+2)]: Sext(12 bits<<1) ➔1111 000000000011

JALR Example: PC← Reg[R2], Reg[R3] ← PC+2 ➔0000 XXX 011 XXX 010

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
K. K. BIRLA Goa Campus, Second Semester 2012-2013
IS F242 Computer Organization, Project, Deadline: 23rd April 2013

Question #3

Design an optimized datapath for the architecture with following instruction formats and instructions. Implement the same as a **Pipelined implementation**. Assume that there is no dependency between instructions in pipeline.

R – type

4 bits	3 bits	3 bits	3 bits	3 bits
Opcode	Shftamt	Rd	Rt	Rs

I – type

4 bits	6 bits	3 bits	3 bits
Opcode	Immediate	Rt/Rd	Rs

J – type

4 bits	12 bits
Opcode	Immediate

Instructions:

ADD Example: Reg[R1] ← Reg[R2] + Reg[R3] ➔0001 XXX 001 011 010

SUB Example: Reg[R1] ← Reg[R2] - Reg[R3] ➔0010 XXX 001 011 010

DIVU Example: Hi, Lo ← Reg[R2] / Reg[R3] ➔0011 XXX XXX 011 010

SLL Example: Reg[R1] ← Reg[R2] << Shftamt (101) ➔0100 101 001 XXX 010

SRL Example: Reg[R1] ← Reg[R2] >> Shftamt (101) ➔0101 101 001 XXX 010

LW Example: Reg[R1] ← Memory[Reg[R2] + Sext(6 bits)] ➔0110 000011 001 010

SW Example: Memory[Reg[R2] + Sext(6 bits)] ← Reg[R1] ➔0111 000011 001 010

BNE Example:NZ flag=1 if Reg[R2] ≠Reg[R1],else NZ flag=0➔1000 000111 001 010

PC ← (PC+2) + Sext(6bits<<1)

ORI Example: Reg[R1] ← Reg[R2] OR Zext(6 bits) ➔1001 000011 001 010

J Example: PC← 3 bits MSB [(PC+2)]: Sext(12 bits<<1) ➔1010 000000000011

JALR Example: PC← Reg[R2], Reg[R3] ← PC+2 ➔1011 XXX 011 XXX 010

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
K. K. BIRLA Goa Campus, Second Semester 2012-2013
IS F242 Computer Organization, Project, Deadline: 23rd April 2013

Question #4

Design an optimized datapath for the architecture with following instruction formats and instructions. Implement the same as a **Single cycle implementation**.

R – type

4 bits	4 bits	4 bits	4 bits
Function field	Rt	Rs and Rd	Opcode

I – type

8 bits	4 bits	4 bits
Immediate	Rs and Rd	Opcode

J – type

12 bits	4 bits
Immediate	Opcode

Instructions:

ADD Example: Reg[R2] ← Reg[R2] + Reg[R3] ➔0001 0011 0010 0000

SUB Example: Reg[R2] ← Reg[R2] - Reg[R3] ➔ 0010 0011 0010 0000

DIVU Example: Hi, Lo ← Reg[R2] / Reg[R3] ➔ 0011 0011 0010 0000

MFHI Example: Reg[R2] ← Hi ➔XXXX XXXX 0010 0001

MFLO Example: Reg[R2] ← Lo ➔ XXXX XXXX 0010 0010

MTLH (Move to Hi and Lo)

Example: Hi ← Reg[R2], Lo← Reg[R3] ➔ XXXX 0011 0010 0011

SLLV Example: Reg[R2] ← Reg[R2] << Reg[R3] ➔ XXXX 0011 0010 0100

SRLV Example: Reg[R2] ← Reg[R2] >> Reg[R3] ➔ XXXX 0011 0010 0101

SRAV Example: Reg[R2] ← Reg[R2] >> Reg[R3] ➔ XXXX 0011 0010 0110

LW Example: Reg[R2] ← Memory[Reg[R2] + Sext(8 bits)] ➔ 00000100 0010 0111

SW Example: Memory[Reg[R2] + Sext(8 bits)] ← Reg[R2] ➔ 00000100 0010 1000

BEQ Example:Z flag=1 if Reg[R2] = Reg[R2+1],else Z flag=0➔00000100 0010 1001

PC ← (PC+2) + Sext(8 bits<<1)

LUI Example: Reg[R3] ← 8 bit immediate:8'd0 ➔10000111 0011 1010

8 bit immediate value is IR[15:8]

ORI Example: Reg[R2] ← Reg[R2] OR Zext(8 bits) ➔11010000 0010 1011

JR Example: PC← Reg[R2] + Sext(8 bits) ➔00000111 0010 1100

JAL Example: PC← 3 bits MSB [(PC+2)]: Sext(12 bits<<1), Reg[R15] ← PC+2
➔ 000000000101 1101

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
K. K. BIRLA Goa Campus, Second Semester 2012-2013
IS F242 Computer Organization, Project, Deadline: 23rd April 2013

Question #5

Design an optimized datapath for the architecture with following instruction formats and instructions. Implement the same as a **Multi cycle implementation**.

R – type

4 bits	4 bits	4 bits	4 bits
Function field	Rt	Rs and Rd	Opcode

I – type

8 bits	4 bits	4 bits
Immediate	Rs and Rd	Opcode

J – type

12 bits	4 bits
Immediate	Opcode

Instructions:

ADD Example: Reg[R2] ← Reg[R2] + Reg[R3] ➔0001 0011 0010 0000

SUB Example: Reg[R2] ← Reg[R2] - Reg[R3] ➔ 0010 0011 0010 0000

DIVU Example: Hi, Lo ← Reg[R2] / Reg[R3] ➔ 0011 0011 0010 0000

MFHI Example: Reg[R2] ← Hi ➔XXXX XXXX 0010 0001

MFLO Example: Reg[R2] ← Lo ➔ XXXX XXXX 0010 0010

MTLH (Move to Hi and Lo)

Example: Hi ← Reg[R2], Lo← Reg[R3] ➔ XXXX 0011 0010 0011

SLLV Example: Reg[R2] ← Reg[R2] << Reg[R3] ➔ XXXX 0011 0010 0100

SRLV Example: Reg[R2] ← Reg[R2] >> Reg[R3] ➔ XXXX 0011 0010 0101

SRAV Example: Reg[R2] ← Reg[R2] >> Reg[R3] ➔ XXXX 0011 0010 0110

LW Example: Reg[R2] ← Memory[Reg[R2] + Sext(8 bits)] ➔ 00000100 0010 0111

SW Example: Memory[Reg[R2] + Sext(8 bits)] ← Reg[R2] ➔ 00000100 0010 1000

BEQ Example:Z flag=1 if Reg[R2] = Reg[R2+1],else Z flag=0➔00000100 0010 1001

PC ← (PC+2) + Sext(8 bits<<1)

LUI Example: Reg[R3] ← 8 bit immediate:8'd0 ➔10000111 0011 1010

8 bit immediate value is IR[15:8]

ORI Example: Reg[R2] ← Reg[R2] OR Zext(8 bits) ➔11010000 0010 1011

JR Example: PC← Reg[R2] + Sext(8 bits) ➔00000111 0010 1100

JAL Example: PC← 3 bits MSB [(PC+2)]: Sext(12 bits<<1), Reg[R15] ← PC+2
➔ 000000000101 1101

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
K. K. BIRLA Goa Campus, Second Semester 2012-2013
IS F242 Computer Organization, Project, Deadline: 23rd April 2013

Question #6

Design an optimized datapath for the architecture with following instruction formats and instructions. Implement the same as a **Pipelined implementation**.

R – type

4 bits	4 bits	4 bits	4 bits
Function field	Rt	Rs and Rd	Opcode

I – type

8 bits	4 bits	4 bits
Immediate	Rs and Rd	Opcode

J – type

12 bits	4 bits
Immediate	Opcode

Instructions:

ADD Example: Reg[R2] ← Reg[R2] + Reg[R3] ➔0001 0011 0010 0000

SUB Example: Reg[R2] ← Reg[R2] - Reg[R3] ➔ 0010 0011 0010 0000

DIVU Example: Hi, Lo ← Reg[R2] / Reg[R3] ➔ 0011 0011 0010 0000

SLLV Example: Reg[R2] ← Reg[R2] << Reg[R3] ➔ XXXX 0011 0010 0001

SRLV Example: Reg[R2] ← Reg[R2] >> Reg[R3] ➔ XXXX 0011 0010 0010

LW Example: Reg[R2] ← Memory[Reg[R2] + Sext(8 bits)] ➔ 00000100 0010 0011

SW Example: Memory[Reg[R2] + Sext(8 bits)] ← Reg[R2] ➔ 00000100 0010 0100

BEQ Example:Z flag=1 if Reg[R2] = Reg[R2+1],else Z flag=0➔00000100 0010 0101

PC ← (PC+2) + Sext(8 bits<<1)

ORI Example: Reg[R2] ← Reg[R2] OR Zext(8 bits) ➔11010000 0010 0110

JR Example: PC← Reg[R2] + Sext(8 bits) ➔00000111 0010 0111

JAL Example: PC← 3 bits MSB [(PC+2)]: Sext(12 bits<<1), Reg[R15] ← PC+2
➔ 000000000101 1000

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
K. K. BIRLA Goa Campus, Second Semester 2012-2013
IS F242 Computer Organization, Project, Deadline: 23rd April 2013

Question #7

Design an optimized datapath for the architecture with following instruction formats and instructions. Implement the same as a **Single cycle implementation**.

R – type

3 bits	3 bits	3 bits	3 bits	4 bits
Shftamt	Function field	Rt	Rs and Rd	Opcode

I – type

6 bits	3 bits	3 bits	4 bits
Immediate	Rt	Rd/Rs	Opcode

J – type

9 bits	3 bits	4 bits
Immediate	Rd	Opcode

Instructions:

ADD Example: Reg[R2] ← Reg[R2] + Reg[R3] ➔ XXX 001 011 010 0000

SUB Example: Reg[R2] ← Reg[R2] - Reg[R3] ➔ XXX 010 011 010 0000

DIVU Example: Hi, Lo ← Reg[R2] / Reg[R3] ➔ XXX 011 011 010 0000

MFHI Example: Reg[R2] ← Hi ➔XXX XXX XXX 010 0001

MFLO Example: Reg[R2] ← Lo ➔ XXX XXX XXX 010 0010

MTLH (Move to Hi and Lo)

Example: Hi ← Reg[R2], Lo← Reg[R3] ➔ XXX XXX 011 010 0011

SLL Example: Reg[R2] ← Reg[R3] << IR[3:0] ➔ 0100 XX 011 010 0100

SRL Example: Reg[R2] ← Reg[R2] >> IR[3:0] ➔ 0100 XX 011 010 0101

SRA Example: Reg[R2] ← Reg[R2] >> IR[3:0] ➔ 0100 XX 011 010 0110

LW Example: Reg[R2] ← Memory[Reg[R3] + Sext(6 bits)] ➔ 000001 011 010 0111

SW Example: Memory[Reg[R3] + Sext(6 bits)] ← Reg[R2] ➔ 000001 011 010 1000

BNE Example:NZ flag=1 if Reg[R2] ≠ Reg[R3],else NZ flag=0➔ 000001 011 010 1001

PC ← (PC+2) + Sext(6 bits<<1)

LUI Example: Reg[R3] ← 8 bit immediate:8'd0 ➔10000111 X 011 1010

8 bit immediate value is IR[15:8]

ORI Example: Reg[R2] ← Reg[R3] OR Zext(6 bits) ➔110100 011 010 1011

JR Example: PC← Reg[R2] + Sext(9 bits) ➔000000111 010 1100

JAL Example: PC← 6 bits MSB [(PC+2)]: Sext(9 bits<<1), Reg[R2] ← PC+2
➔ 000000111 010 1101

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
K. K. BIRLA Goa Campus, Second Semester 2012-2013
IS F242 Computer Organization, Project, Deadline: 23rd April 2013

Question #8

Design an optimized datapath for the architecture with following instruction formats and instructions. Implement the same as a **Multi cycle implementation**.

R – type

3 bits	3 bits	3 bits	3 bits	4 bits
Shftamt	Function field	Rt	Rs and Rd	Opcode

I – type

6 bits	3 bits	3 bits	4 bits
Immediate	Rt	Rd/Rs	Opcode

J – type

9 bits	3 bits	4 bits
Immediate	Rd	Opcode

Instructions:

ADD Example: Reg[R2] ← Reg[R2] + Reg[R3] ➔ XXX 001 011 010 0000

SUB Example: Reg[R2] ← Reg[R2] - Reg[R3] ➔ XXX 010 011 010 0000

DIVU Example: Hi, Lo ← Reg[R2] / Reg[R3] ➔ XXX 011 011 010 0000

MFHI Example: Reg[R2] ← Hi ➔XXX XXX XXX 010 0001

MFLO Example: Reg[R2] ← Lo ➔ XXX XXX XXX 010 0010

MTLH (Move to Hi and Lo)

Example: Hi ← Reg[R2], Lo← Reg[R3] ➔ XXX XXX 011 010 0011

SLL Example: Reg[R2] ← Reg[R3] << IR[3:0] ➔ 0100 XX 011 010 0100

SRL Example: Reg[R2] ← Reg[R2] >> IR[3:0] ➔ 0100 XX 011 010 0101

SRA Example: Reg[R2] ← Reg[R2] >> IR[3:0] ➔ 0100 XX 011 010 0110

LW Example: Reg[R2] ← Memory[Reg[R3] + Sext(6 bits)] ➔ 000001 011 010 0111

SW Example: Memory[Reg[R3] + Sext(6 bits)] ← Reg[R2] ➔ 000001 011 010 1000

BNE Example:NZ flag=1 if Reg[R2] ≠ Reg[R3],else NZ flag=0➔ 000001 011 010 1001

PC ← (PC+2) + Sext(6 bits<<1)

LUI Example: Reg[R3] ← 8 bit immediate:8'd0 ➔10000111 X 011 1010

8 bit immediate value is IR[15:8]

ORI Example: Reg[R2] ← Reg[R3] OR Zext(6 bits) ➔110100 011 010 1011

JR Example: PC← Reg[R2] + Sext(9 bits) ➔000000111 010 1100

JAL Example: PC← 6 bits MSB [(PC+2)]: Sext(9 bits<<1), Reg[R2] ← PC+2
➔ 000000111 010 1101

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
K. K. BIRLA Goa Campus, Second Semester 2012-2013
IS F242 Computer Organization, Project, Deadline: 23rd April 2013

Question #9

Design an optimized datapath for the architecture with following instruction formats and instructions. Implement the same as a **Pipelined implementation**.

R – type

3 bits	3 bits	3 bits	3 bits	4 bits
Shftamt	Function field	Rt	Rs and Rd	Opcode

I – type

6 bits	3 bits	3 bits	4 bits
Immediate	Rt	Rd/Rs	Opcode

J – type

9 bits	3 bits	4 bits
Immediate	Rd	Opcode

Instructions:

ADD Example: Reg[R2] ← Reg[R2] + Reg[R3] ➔ XXX 001 011 010 0000

SUB Example: Reg[R2] ← Reg[R2] - Reg[R3] ➔ XXX 010 011 010 0000

DIVU Example: Hi, Lo ← Reg[R2] / Reg[R3] ➔ XXX 011 011 010 0000

SLL Example: Reg[R2] ← Reg[R3] << IR[3:0] ➔ 0100 XX 011 010 0001

SRL Example: Reg[R2] ← Reg[R2] >> IR[3:0] ➔ 0100 XX 011 010 0010

LW Example: Reg[R2] ← Memory[Reg[R3] + Sext(6 bits)] ➔ 000001 011 010 0011

SW Example: Memory[Reg[R3] + Sext(6 bits)] ← Reg[R2] ➔ 000001 011 010 0100

BNE Example:NZ flag=1 if Reg[R2] ≠ Reg[R3],else NZ flag=0➔ 000001 011 010 0101

PC ← (PC+2) + Sext(6 bits<<1)

ORI Example: Reg[R2] ← Reg[R3] OR Zext(6 bits) ➔110100 011 010 0110

JR Example: PC← Reg[R2] + Sext(9 bits) ➔000000111 010 0111

JAL Example: PC← 6 bits MSB [(PC+2)]: Sext(9 bits<<1), Reg[R2] ← PC+2
➔ 000000111 010 1000

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
K. K. BIRLA Goa Campus, Second Semester 2012-2013
IS F242 Computer Organization, Project, Deadline: 23rd April 2013

Question #10

Design an optimized datapath for the architecture with following instruction formats and instructions. Implement the same as a **Single cycle implementation**.

R – type

4 bits	3 bits	3 bits	3 bits	3 bits
Opcode	Rd	Rt	Rs1	Rs2

I – type

4 bits	3 bits	3 bits	6 bits
Opcode	Rd/Rt	Rs	Immediate

J – type

4 bits	12 bits
Opcode	Immediate

Instructions:

ADDD Example: Reg[R2] ← Reg[R1] + Reg[R3] + Reg[R4]➔ 0000 010 100 001 011

SUBB Example: Reg[R2] ← Reg[R1] - Reg[R3] - Reg[R4]➔ 0001 010 100 001 011

ADSUB Example: Reg[R2] ← Reg[R1] + Reg[R3] - Reg[R4]➔ 0010 010 100 001 011

DIVU Example: Reg[R2],Reg[R4] ← Reg[R1] / Reg[R3] ➔ 0011 010 100 001 011

ADDI Example: Reg[R2] ← Reg[R1] + Reg[R4] + Sext(3 bits)➔ 0100 010 100 001 011

SLL Example: Reg[R2] ← Reg[R3] << 6 bits Imm ➔ 0101 010 011 000111

SRL Example: Reg[R2] ← Reg[R3] >> 6 bits Imm ➔ 0110 010 011 000111

SRA Example: Reg[R2] ← Reg[R3] >> 6 bits Imm ➔ 0111 010 011 000111

LW Example: Reg[R2] ← Memory[Reg[R3] + Sext(6 bits)] ➔ 1000 010 011 000001

SW Example: Memory[Reg[R3] + Sext(6 bits)] ← Reg[R2] ➔ 1001 010 011 000001

BNE Example:NZ flag=1 if Reg[R2] ≠ Reg[R3],else NZ flag=0➔ 1010 011 010 000011

PC ← (PC+2) + Sext(6 bits<<1)

ORI Example: Reg[R2] ← Reg[R3] OR Zext(6 bits) ➔1011 010 011 110100

J Example: PC← 3 bits MSB [(PC+2)]: Sext(12 bits<<1) ➔1100 000000111010

JALR Example: PC← Reg[R2] + Reg[R3], Reg[R4]← PC+2➔1101 100 XXX 010 011

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
K. K. BIRLA Goa Campus, Second Semester 2012-2013
IS F242 Computer Organization, Project, Deadline: 23rd April 2013

Question #11

Design an optimized datapath for the architecture with following instruction formats and instructions. Implement the same as a **Multi cycle implementation**.

R – type

4 bits	3 bits	3 bits	3 bits	3 bits
Opcode	Rd	Rt	Rs1	Rs2

I – type

4 bits	3 bits	3 bits	6 bits
Opcode	Rd/Rt	Rs	Immediate

J – type

4 bits	12 bits
Opcode	Immediate

Instructions:

ADDD Example: Reg[R2] ← Reg[R1] + Reg[R3] + Reg[R4]➔ 0000 010 100 001 011

SUBB Example: Reg[R2] ← Reg[R1] - Reg[R3] - Reg[R4]➔ 0001 010 100 001 011

ADSUB Example: Reg[R2] ← Reg[R1] + Reg[R3] - Reg[R4]➔ 0010 010 100 001 011

DIVU Example: Reg[R2],Reg[R4] ← Reg[R1] / Reg[R3] ➔ 0011 010 100 001 011

ADDI Example: Reg[R2] ← Reg[R1] + Reg[R4] + Sext(3 bits)➔ 0100 010 100 001 011

SLL Example: Reg[R2] ← Reg[R3] << 6 bits Imm ➔ 0101 010 011 000111

SRL Example: Reg[R2] ← Reg[R3] >> 6 bits Imm ➔ 0110 010 011 000111

SRA Example: Reg[R2] ← Reg[R3] >> 6 bits Imm ➔ 0111 010 011 000111

LW Example: Reg[R2] ← Memory[Reg[R3] + Sext(6 bits)] ➔ 1000 010 011 000001

SW Example: Memory[Reg[R3] + Sext(6 bits)] ← Reg[R2] ➔ 1001 010 011 000001

BNE Example:NZ flag=1 if Reg[R2] ≠ Reg[R3],else NZ flag=0➔ 1010 011 010 000011

PC ← (PC+2) + Sext(6 bits<<1)

ORI Example: Reg[R2] ← Reg[R3] OR Zext(6 bits) ➔1011 010 011 110100

J Example: PC← 3 bits MSB [(PC+2)]: Sext(12 bits<<1) ➔1100 000000111010

JALR Example: PC← Reg[R2] + Reg[R3], Reg[R4]← PC+2➔1101 100 XXX 010 011

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
K. K. BIRLA Goa Campus, Second Semester 2012-2013
IS F242 Computer Organization, Project, Deadline: 23rd April 2013

Question #12

Design an optimized datapath for the architecture with following instruction formats and instructions. Implement the same as a **Pipelined implementation**.

R – type

4 bits	3 bits	3 bits	3 bits	3 bits
Opcode	Rd	Rt	Rs1	Rs2

I – type

4 bits	3 bits	3 bits	6 bits
Opcode	Rd/Rt	Rs	Immediate

J – type

4 bits	12 bits
Opcode	Immediate

Instructions:

ADDD Example: Reg[R2] ← Reg[R1] + Reg[R3] + Reg[R4]➔ 0000 010 100 001 011

SUBB Example: Reg[R2] ← Reg[R1] - Reg[R3] - Reg[R4]➔ 0001 010 100 001 011

ADSUB Example: Reg[R2] ← Reg[R1] + Reg[R3] - Reg[R4]➔ 0010 010 100 001 011

ADDI Example: Reg[R2] ← Reg[R1] + Reg[R4] + Sext(3 bits)➔ 0011 010 100 001 011

SLL Example: Reg[R2] ← Reg[R3] << 6 bits Imm ➔ 0100 010 011 000111

SRL Example: Reg[R2] ← Reg[R3] >> 6 bits Imm ➔ 0101 010 011 000111

LW Example: Reg[R2] ← Memory[Reg[R3] + Sext(6 bits)] ➔ 0110 010 011 000001

SW Example: Memory[Reg[R3] + Sext(6 bits)] ← Reg[R2] ➔ 0111 010 011 000001

BNE Example:NZ flag=1 if Reg[R2] ≠ Reg[R3],else NZ flag=0➔ 1000 011 010 000011

PC ← (PC+2) + Sext(6 bits<<1)

J Example: PC← 3 bits MSB [(PC+2)]: Sext(12 bits<<1) ➔1001 000000111010

JALR Example: PC← Reg[R2] + Reg[R3], Reg[R4]← PC+2➔1010 100 XXX 010 011