

ICE-AH ARCHITECTURE

Hoda Ragaie
Rana Emad
Fady Aziz
Sherif Sameh
Nour Abu-zeid



BASIC INSTRUCTION FORMAT

- Each instruction is of size 32-bits.
- The R-Type immediate is neglected.
- Instruction format supports both positive and negative numbers.

R-TYPE

OPCODE	RS	RT	RD	IMMEDIATE
31->28	27->23	22->18	17->13	12->0

I-TYPE

OPCODE	RS	RT	IMMEDIATE
31->28	27->23	22->18	17->0

J-TYPE

OPCODE	JUMP-ADDRESS
31->28	27->0

COMPONENTS



- 1. Cache : size 8.**
- 2. Data Memory : size 1024 - Integers only (Word addressing).**
- 3. Instruction Memory : size 1024 (Word addressing).**
- 4. Program Counter : initially zero.**
- 5. Register : has a name, description and an integer value.
{ “reg0”, “reg1”, “reg2”, “reg31” } all are general purpose registers.**
- 6. Register File : consists of 32 registers**
- 7. Pipeline : consists of 4 pipelines.
{ “IF-ID”, “ID-EX”, “EX-MA”, “MA-WR” }**



STAGES



By order we have:

- 1. Write Back**
- 2. Memory Access**
- 3. Execute**
- 4. Instruction Decode**
- 5. Instruction Fetch**

WRITE BACK

- if MA_WB is empty, the register file remains unchanged.
- If clock cycle = WritebackClk, we get the values from pipeline (MA_WB) to write them back to register file according to the following cases:
 - A. If the instruction is an R-Type instruction (add/sub/mult/slt/and) , the data is written back to the rd register.
 - B. If the instruction is an I-Type (addi/ori/sll/srl) , the data is written back to the rt register.
 - C. For any other instructions, we do not need to write back.

MEMORY ACCESS

- When EX_MA is empty, we do not access the cache.
- If, clock cycle = MemClk, we access the cache according to the following cases:
 - A. If the instruction was sw, we write the value of rt register in the address at index: rs + sign extend.
If the address entered was not in the cache, it is written in the corresponding address in the data memory and written to the cache with the updated value. Otherwise, we update the existing value in cache and data memory with the new data.
 - B. If the instruction was LW, we read the value currently at index (rs + sign extend) from the cache if exists else, we fetch it from data memory, write it in the cache. The value read is then passed to the write back so that in the next cycle, the rt register is updated with its value.
 - C. If the instruction was branch or jump, I set the PC with the new address; the IF_ID , ID_EX , EX_MA are cleared.

CONTROL UNIT

- The opcode is the input to the control unit and according to it the control signals are set.

	Set to 1
add/sub/mult/SLT/AND	regDst, regWrite
BNQ/BGT	Branch
SW	MemWrite, lslmmediate
LW	MemRead, MemtoReg, RegWrite, lslmmediate
Jump	Jump
Addi/Ori/SLL/SRL	lslmmediate, RegWrite

Control Signals	<i>If not set</i>	<i>If set</i>
regDst	Destination register comes from rt field.	Destination register comes from the rd field.
Branch	PC ->PC + 1	PC ->Branch Address
MemRead	None.	Contents of address input are copied to Read Data.
MemToReg	Value of register Write Data is from ALU	Value of register Write Data is memory Read Data.
MemWrite	None.	Write Data is written to Address
lslmmediate	Second ALU operand is Read Data 2.	Second ALU operand is immediate field.
RegWrite	None	Write register is written to with Write Data.
Jump	PC ->PC + 1	PC ->Jump Address

ALU

- The opcode is the input to the ALU and according to it the ALU controls are set.

<u>ALU Controls</u>	<u>Operation</u>
“0000”	Add
“0001”	Sub
“0010”	mult
“0110”	SRL
“0100”	AND
“0011”	OR
“0111”	SLT
“0101”	SLL
“1000”	SBT
“1001”	Jump

EXECUTE

- if EX_MA is empty, the register file remains unchanged.
If clock cycle = ExecuteClk, we get the values from pipeline (EX_MA)
- The operation is determined according to the ALU Control.
- In case of jump (“1001”), the jump address is passed to Memory Access through Pipeline.
- For BNQ, if the subtraction result is not equal 0, the branch address is set to the sign extend.
- For BGT, if the sbt (set bigger than) result is one, the branch address is set to the sign extend.
- For SRL, I shift the rs value by the signExtend to the right. And same approach for SLL except shifting is to the left.
- The ZeroFlag and The OneFlag are set according to the result.

DECODE

- We decode according to the instruction format previously mentioned in slide 2.
- The opcode is passed to the control unit.

OPCode	Instruction Type
“0000”	add
“0001”	Addi
“0010”	sub
“0011”	Mult
“0100”	AND
“0101”	Ori
“0110”	SLL
“0111”	SRL
“1000”	Lw
“1001”	Sw
“1010”	BNQ
“1011”	BGT
“1100”	SLT
“1101”	Jump

FETCH

- From the Program Counter, we get the current PC and fetch the instruction at index = PC.
- PC is incremented by 1 for the next instruction fetch and passed to the IF_ID pipeline.
- The clocks :
 1. decodeClk
 2. executeClk
 3. memClk
 4. writebackClkare calculated and passed to the IF_ID pipeline.

IF_ID

PC

Instruction

fetchCLK

executeCLK

memCLK

decodeCLK

writebackCLK

exitFlagFetch

ID_EX

opCode
JumpAddress
immediate
rtValue
rsValue
rdIndex
rtIndex
signExtend
regDst
branch
jump
memRead
memWrite
memToReg

regWrite
isImmediate
ALU control
exitFlagDecode

EX_MA

OneFlag

ALU result

ZeroFlag

canJump

Branch Address

canBranch

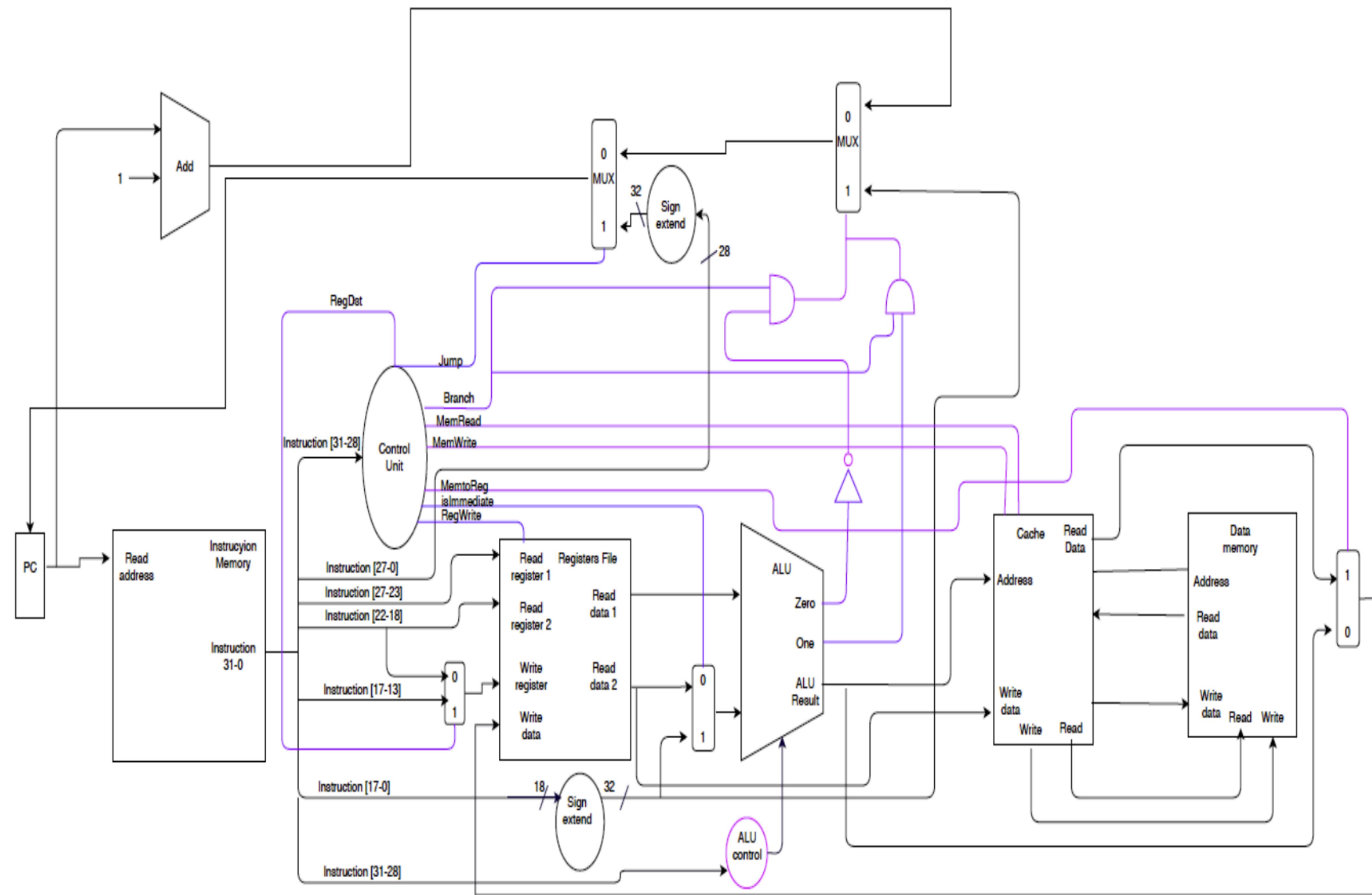
Jump Address

exitFlagExecute

MA_WB

Memory output

exitFlagMemory



TEST CASES

ClockCycle: 0

MA_WB{}
MA_WB{}
EX_MA{}
ID_EX{}

Instruction Fetching ... at Clock Cycle: 0

IF_ID{writebackCLK=4, memCLK=3, Instruction=101000000000100000000000000011, fetchCLK=0, PC=1, decodeCLK=1, executeCLK=2}

ClockCycle: 1

MA_WB{}
MA_WB{}
EX_MA{}

Instruction Decode ... at ClockCycle: 1

ID_EX{Instruction=1010000000001000000000000000011, memRead=0, JumpAddress=000000000000100000000000000011, fetchCLK=0, branch=1, immediate=00000000000000000000000000000011, gWrite=0, rsValue=1, jump=0, memCLK=3, rtValue=2, executeCLK=2, ALUControl=0001, memToReg=0, writebackCLK=4, PC=1, decodeCLK=1, memWrite=0, isImmediate=0, rdIndex=0, regDst=0, rtIndex=1, opCode=1010}

Instruction Fetching ... at Clock Cycle: 1

IF_ID{writebackCLK=5, memCLK=4, Instruction=000001000101001100000000000000, fetchCLK=1, PC=2, decodeCLK=2, executeCLK=3}

ClockCycle: 2

MA_WB{}
MA_WB{}

Execute at ... ClockCycle: 2

EX_MA{Instruction=1010000000001000000000000000011, JumpAddress=000000000000100000000000000011, memRead=0, fetchCLK=0, branch=1, immediate=00000000000000000000000000000011, regWrite=0, rsValue=1, jump=0, memCLK=3, canJump=false, rtValue=2, zeroFlag=false, jumpAddress=-1, executeCLK=2, ALUControl=0001, oneFlag=true, ALUresult=-1, memToReg=0, writebackCLK=4, PC=1, decodeCLK=1, memWrite=0, isImmediate=0, rdIndex=0, regDst=0, rtIndex=1, canBranch=true, branchAddress=3, opCode=1010}

Instruction Decode ... at ClockCycle: 2

ID_EX{Instruction=000001000101001100000000000000, memRead=0, JumpAddress=111100100010100110000000000000, fetchCLK=1, branch=0, immediate=111111111111100110000000000000, regWrite=1, rsValue=5, jump=0, memCLK=4, rtValue=6, executeCLK=3, ALUControl=0000, memToReg=0, writebackCLK=5, PC=2, decodeCLK=2, memWrite=0, isImmediate=0, rdIndex=6, regDst=1, rtIndex=5, opCode=0000}

Instruction Fetching ... at Clock Cycle: 2

IF_ID{writebackCLK=6, memCLK=5, Instruction=1101000000000000000000000000101, fetchCLK=2, PC=3, decodeCLK=3, executeCLK=4}

ClockCycle: 3

MA_WB{}

Memory Access at ... ClockCycle: 3

No data to output from memory; e.g sw

MA_WB{Instruction=1010000000001000000000000000011, JumpAddress=000000000000100000000000000011, memRead=0, fetchCLK=0, branch=1, immediate=00000000000000000000000000000011, egWrite=0, rsValue=1, jump=0, memCLK=3, canJump=false, rtValue=2, zeroFlag=false, jumpAddress=-1, executeCLK=2, ALUControl=0001, memoryOutput=-1, oneFlag=true, memToReg=0, ALUresult=-1, writebackCLK=4, PC=1, decodeCLK=1, memWrite=0, isImmediate=0, rdIndex=0, regDst=0, rtIndex=1, opCode=1010, branchAddress=3, canBranch=true}

EX_MA{}
ID_EX{}

Instruction Fetching ... at Clock Cycle: 3

IF_ID{writebackCLK=7, memCLK=6, Instruction=010100111010000000000000000011, fetchCLK=3, PC=4, decodeCLK=4, executeCLK=5}

ClockCycle: 4

WriteBack at ... *ClockCycle: 4*

There was no data to write back

MA_WB{*Instruction*=1010000000001000000000000000011, *JumpAddress*=000000000001000000000000000011, *memRead*=0, *fetchCLK*=0, *branch*=1, *immediate*=00000000000000000000000000000011, *egWrite*=0, *rsValue*=1, *jump*=0, *memCLK*=3, *canJump*=false, *rtValue*=2, *zeroFlag*=false, *jumpAddress*=-1, *executeCLK*=2, *ALUControl*=001, *memoryOutput*=-1, *oneFlag*=true, *memToReg*=0, *ALUresult*=-1, *writebackCLK*=4, *PC*=1, *decodeCLK*=1, *memWrite*=0, *isImmediate*=0, *rdIndex*=0, *regDst*=0, *rtIndex*=1, *opCode*=1010, *branchAddress*=3, *canBranch*=true}

MA_WB{*Instruction*=10100000000010000000000000000011, *JumpAddress*=000000000001000000000000000011, *memRead*=0, *fetchCLK*=0, *branch*=1, *immediate*=00000000000000000000000000000011, *regWrite*=0, *rsValue*=1, *jump*=0, *memCLK*=3, *canJump*=false, *rtValue*=2, *zeroFlag*=false, *jumpAddress*=-1, *executeCLK*=2, *ALUControl*=001, *memoryOutput*=-1, *oneFlag*=true, *memToReg*=0, *LUresult*=-1, *writebackCLK*=4, *PC*=1, *decodeCLK*=1, *memWrite*=0, *isImmediate*=0, *rdIndex*=0, *regDst*=0, *rtIndex*=1, *opCode*=1010, *branchAddress*=3, *canBranch*=true}

EX_MA{}

Instruction Decode ... at *ClockCycle: 4*

ID_EX{*Instruction*=01010011101000000000000000000011, *memRead*=0, *JumpAddress*=00000111010000000000000000000011, *fetchCLK*=3, *branch*=0, *immediate*=00000000000000000000000000000011, *egWrite*=1, *rsValue*=8, *jump*=0, *memCLK*=6, *rtValue*=9, *executeCLK*=5, *ALUControl*=0011, *memToReg*=0, *writebackCLK*=7, *PC*=4, *decodeCLK*=4, *memWrite*=0, *isImmediate*=1, *rdIndex*=0, *regDst*=0, *rtIndex*=8, *opCode*=0101}

Instruction Fetching ... at *Clock Cycle: 4*

IF_ID{*writebackCLK*=8, *memCLK*=7, *Instruction*=00010101101001111111111111110, *fetchCLK*=4, *PC*=5, *decodeCLK*=5, *executeCLK*=6}

ClockCycle: 5

WriteBack at ... *ClockCycle: 5*

There was no data to write back

MA_WB{*Instruction*=1010000000001000000000000000011, *JumpAddress*=000000000001000000000000000011, *memRead*=0, *fetchCLK*=0, *branch*=1, *immediate*=00000000000000000000000000000011, *gWrite*=0, *rsValue*=1, *jump*=0, *memCLK*=3, *canJump*=false, *rtValue*=2, *zeroFlag*=false, *jumpAddress*=-1, *executeCLK*=2, *ALUControl*=001, *memoryOutput*=-1, *oneFlag*=true, *memToReg*=0, *ALUresult*=-1, *writebackCLK*=4, *PC*=1, *decodeCLK*=1, *memWrite*=0, *isImmediate*=0, *rdIndex*=0, *regDst*=0, *rtIndex*=1, *opCode*=1010, *branchAddress*=3, *canBranch*=true}

MA_WB{*Instruction*=1010000000001000000000000000011, *JumpAddress*=000000000001000000000000000011, *memRead*=0, *fetchCLK*=0, *branch*=1, *immediate*=00000000000000000000000000000011, *regWrite*=0, *rsValue*=1, *jump*=0, *memCLK*=3, *canJump*=false, *rtValue*=2, *zeroFlag*=false, *jumpAddress*=-1, *executeCLK*=2, *ALUControl*=001, *memoryOutput*=-1, *oneFlag*=true, *memToReg*=0, *ALUresult*=-1, *writebackCLK*=4, *PC*=1, *decodeCLK*=1, *memWrite*=0, *isImmediate*=0, *rdIndex*=0, *regDst*=0, *rtIndex*=1, *opCode*=1010, *branchAddress*=3, *canBranch*=true}

Execute at ... *ClockCycle: 5*

EX_MA{*Instruction*=01010011101000000000000000000011, *JumpAddress*=000001110100000000000000000011, *memRead*=0, *fetchCLK*=3, *immediate*=00000000000000000000000000000011, *branch*=0, *egWrite*=1, *rsValue*=8, *jump*=0, *memCLK*=6, *canJump*=false, *rtValue*=9, *zeroFlag*=false, *jumpAddress*=-1, *executeCLK*=5, *ALUControl*=0011, *oneFlag*=true, *ALUresult*=11, *memToReg*=0, *writebackCLK*=7, *PC*=4, *decodeCLK*=4, *memWrite*=0, *isImmediate*=1, *rdIndex*=0, *regDst*=0, *rtIndex*=8, *canBranch*=false, *branchAddress*=0, *opCode*=0101}

Instruction Decode ... at *ClockCycle: 5*

ID_EX{*Instruction*=00010101101001111111111111110, *memRead*=0, *JumpAddress*=11110101101100111111111111110, *fetchCLK*=4, *branch*=0, *immediate*=1111111111111111111111110, *regWrite*=1, *rsValue*=12, *jump*=0, *memCLK*=7, *rtValue*=13, *executeCLK*=6, *ALUControl*=0000, *memToReg*=0, *writebackCLK*=8, *PC*=5, *decodeCLK*=5, *memWrite*=0, *isImmediate*=1, *rdIndex*=31, *regDst*=0, *rtIndex*=12, *opCode*=0001}

Instruction Fetching ... at *Clock Cycle: 5*

IF_ID{*writebackCLK*=9, *memCLK*=8, *Instruction*=001001110011010111100000000000, *fetchCLK*=5, *PC*=6, *decodeCLK*=6, *executeCLK*=7}

ClockCycle: 6

WriteBack at ... *ClockCycle: 6*

No data to write back

```
MA_WB{Instruction=10100000000100000000000000000011, JumpAddress=00000000000100000000000000000011, memRead=0, fetchCLK=0, branch=1, immediate=0000000000000000000000000000000011, regWrite=0, rsValue=1, jump=0, memCLK=3, canJump=false, rtValue=2, zeroFlag=false, jumpAddress=-1, executeCLK=2, ALUControl=0001, memoryOutput=-1, oneFlag=true, memToReg=0, ALUresult=-1, writebackCLK=4, PC=1, decodeCLK=1, memWrite=0, isImmediate=0, rdIndex=0, regDst=0, rtIndex=1, opCode=1010, branchAddress=3, canBranch=true}
```

Memory Access at ... *ClockCycle: 6*

No data to output from memory; e.g. sw

```
MA_WB{Instruction=01010011101000000000000000000011, memRead=0, JumpAddress=00000111010000000000000000000011, fetchCLK=3, branch=0, immediate=0000000000000000000000000000000011, regWrite=1, rsValue=8, jump=0, memCLK=6, canJump=false, rtValue=9, zeroFlag=false, executeCLK=5, jumpAddress=-1, ALUControl=0011, memoryOutput=-1, oneFlag=true, memToReg=0, ALUresult=11, writebackCLK=7, PC=4, decodeCLK=4, memWrite=0, isImmediate=1, rdIndex=0, regDst=0, rtIndex=8, opCode=0101, branchAddress=0, canBranch=false}
```

Execute at ... *ClockCycle: 6*

```
EX_MA{Instruction=0001010110100111111111111110, JumpAddress=111101011011001111111111111110, memRead=0, fetchCLK=4, immediate=111111111111111111111111111110, branch=0, regWrite=1, rsValue=12, jump=0, memCLK=7, canJump=false, rtValue=13, zeroFlag=false, executeCLK=6, ALUControl=0000, oneFlag=true, ALUresult=10, memToReg=0, writebackCLK=8, PC=5, decodeCLK=5, memWrite=0, isImmediate=1, rdIndex=31, regDst=0, rtIndex=12, canBranch=false, branchAddress=0, opCode=0001}
```

Instruction Decode ... at *ClockCycle: 6*

```
ID_EX{Instruction=00100111001101011110000000000000, memRead=0, JumpAddress=11110111001101011110000000000000, fetchCLK=5, branch=0, immediate=111111111111011110000000000000, regWrite=1, rsValue=15, jump=0, memCLK=8, rtValue=14, executeCLK=7, ALUControl=0001, memToReg=0, writebackCLK=9, PC=6, decodeCLK=6, memWrite=0, isImmediate=0, rdIndex=15, regDst=1, rtIndex=13, opCode=0010}
```

Instruction Fetching ... at *Clock Cycle: 6*

```
IF_ID{writebackCLK=10, memCLK=9, Instruction=1000010010101000000000000000011, fetchCLK=6, PC=7, decodeCLK=7, executeCLK=8}
```

ClockCycle: 7

WriteBack at ... *ClockCycle: 7*

```
MA_WB{Instruction=01010011101000000000000000000011, memRead=0, JumpAddress=00000111010000000000000000000011, fetchCLK=3, branch=0, immediate=0000000000000000000000000000000011, regWrite=1, rsValue=8, jump=0, memCLK=6, canJump=false, rtValue=9, zeroFlag=false, executeCLK=5, jumpAddress=-1, ALUControl=0011, memoryOutput=-1, oneFlag=true, memToReg=0, ALUresult=11, writebackCLK=7, PC=4, decodeCLK=4, memWrite=0, isImmediate=1, rdIndex=0, regDst=0, rtIndex=8, opCode=0101, branchAddress=0, canBranch=false}
```

Memory Access at ... *ClockCycle: 7*

No data to output from memory; e.g. sw

```
MA_WB{Instruction=00010101101001111111111111110, memRead=0, JumpAddress=11110101101100111111111111110, fetchCLK=4, branch=0, immediate=11111111111111111111111111110, regWrite=1, rsValue=12, jump=0, memCLK=7, canJump=false, rtValue=13, zeroFlag=false, executeCLK=6, jumpAddress=-1, ALUControl=0000, memoryOutput=-1, oneFlag=true, memToReg=0, ALUresult=10, writebackCLK=8, PC=5, decodeCLK=5, memWrite=0, isImmediate=1, rdIndex=31, regDst=0, rtIndex=12, opCode=0001, branchAddress=0, canBranch=false}
```

Execute at ... *ClockCycle: 7*

```
EX_MA{Instruction=00100111001101011110000000000000, JumpAddress=11110111001101011110000000000000, memRead=0, fetchCLK=5, immediate=111111111111011110000000000000, branch=0, regWrite=1, rsValue=15, jump=0, memCLK=8, canJump=false, rtValue=14, zeroFlag=false, executeCLK=7, ALUControl=0001, oneFlag=true, ALUresult=1, memToReg=0, writebackCLK=9, PC=6, decodeCLK=6, memWrite=0, isImmediate=0, rdIndex=15, regDst=1, rtIndex=13, canBranch=false, branchAddress=0, opCode=0010}
```

Instruction Decode ... at *ClockCycle: 7*

```
ID_EX{Instruction=10000100101010000000000000000011, memRead=1, JumpAddress=00000100101010000000000000000011, fetchCLK=6, branch=0, immediate=0000000000000000000000000000000011, regWrite=1, rsValue=10, jump=0, memCLK=9, rtValue=11, executeCLK=8, ALUControl=0000, memToReg=1, writebackCLK=10, PC=7, decodeCLK=7, memWrite=0, isImmediate=1, rdIndex=0, regDst=0, rtIndex=10, opCode=1000}
```

Instruction Fetching ... at *Clock Cycle: 7*

```
IF_ID{writebackCLK=11, memCLK=10, Instruction=00111000010001100100000000000000, fetchCLK=7, PC=8, decodeCLK=8, executeCLK=9}
```

ClockCycle: 8

WriteBack at ... ClockCycle: 8
MA_WB{*Instruction*=0001010110110011111111111111110, memRead=0, *JumpAddress*=1111010110110011111111111111110, fetchCLK=4, branch=0, immediate=1111111111111111111111111111110, regWrite=1, rsValue=12, jump=0, memCLK=7, canJump=false, rtValue=13, zeroFlag=false, executeCLK=6, jumpAddress=-1, *ALUControl*=0000, memoryOutput=-1, oneFlag=true, memToReg=0, *ALUresult*=10, writebackCLK=8, PC=5, decodeCLK=5, memWrite=0, isImmediate=1, rdIndex=31, regDst=0, rtIndex=12, opCode=0001, branchAddress=0, canBranch=false}

Memory Access at ... ClockCycle: 8
No data to output from memory; e.g sw
MA_WB{*Instruction*=0010011100110101110000000000000, memRead=0, *JumpAddress*=1111011100110101110000000000000, fetchCLK=5, branch=0, immediate=1111111111101111000000000000000, regWrite=1, rsValue=15, jump=0, memCLK=8, canJump=false, rtValue=14, zeroFlag=false, executeCLK=7, jumpAddress=-1, *ALUControl*=0001, memoryOutput=-1, oneFlag=true, memToReg=0, *ALUresult*=1, writebackCLK=9, PC=6, decodeCLK=6, memWrite=0, isImmediate=0, rdIndex=15, regDst=1, rtIndex=13, opCode=0010, branchAddress=0, canBranch=false}

Execute at ... ClockCycle: 8
EX_MA{*Instruction*=1000010010101000000000000000011, *JumpAddress*=00000100101010000000000000000011, memRead=1, fetchCLK=6, immediate=00000000000000000000000000000011, branch=0, regWrite=1, rsValue=10, jump=0, memCLK=9, canJump=false, rtValue=11, zeroFlag=false, jumpAddress=-1, executeCLK=8, *ALUControl*=0000, oneFlag=true, *ALUresult*=13, memToReg=1, writebackCLK=10, PC=7, decodeCLK=7, memWrite=0, isImmediate=1, rdIndex=0, regDst=0, rtIndex=10, canBranch=false, branchAddress=0, opCode=1000}

Instruction Decode ... at ClockCycle: 8
ID_EX{*Instruction*=0011100001000110010000000000000, memRead=0, *JumpAddress*=0000100001000110010000000000000, fetchCLK=7, branch=0, immediate=0000000000001001000000000000000, regWrite=1, rsValue=17, jump=0, memCLK=10, rtValue=18, executeCLK=9, *ALUControl*=0010, memToReg=0, writebackCLK=11, PC=8, decodeCLK=8, memWrite=0, isImmediate=0, rdIndex=18, regDst=1, rtIndex=17, opCode=0011}

Instruction Fetching ... at Clock Cycle: 8
IF_ID{writebackCLK=12, memCLK=11, *Instruction*=0100100111010010101000000000000, fetchCLK=8, PC=9, decodeCLK=9, executeCLK=10}

ClockCycle: 9

WriteBack at ... ClockCycle: 9
MA_WB{*Instruction*=0010011100110111100000000000000, memRead=0, *JumpAddress*=1111011100110111100000000000000, fetchCLK=5, branch=0, immediate=1111111111101111000000000000000, regWrite=1, rsValue=15, jump=0, memCLK=8, canJump=false, rtValue=14, zeroFlag=false, executeCLK=7, jumpAddress=-1, *ALUControl*=0001, memoryOutput=-1, oneFlag=true, memToReg=0, *ALUresult*=1, writebackCLK=9, PC=6, decodeCLK=6, memWrite=0, isImmediate=0, rdIndex=15, regDst=1, rtIndex=13, opCode=0010, branchAddress=0, canBranch=false}

Memory Access at ... ClockCycle: 9
MA_WB{*Instruction*=10000100101010000000000000000011, memRead=1, *JumpAddress*=00000100101010000000000000000011, fetchCLK=6, branch=0, immediate=00000000000000000000000000000011, regWrite=1, rsValue=10, jump=0, memCLK=9, canJump=false, rtValue=11, zeroFlag=false, executeCLK=8, jumpAddress=-1, *ALUControl*=0000, memoryOutput=14, oneFlag=true, memToReg=1, *ALUresult*=13, writebackCLK=10, PC=7, decodeCLK=7, memWrite=0, isImmediate=1, rdIndex=0, regDst=0, rtIndex=10, opCode=1000, branchAddress=0, canBranch=false}

Execute at ... ClockCycle: 9
EX_MA{*Instruction*=0011100001000110010000000000000, *JumpAddress*=0000100001000110010000000000000, memRead=0, fetchCLK=7, immediate=0000000000001001000000000000000, branch=0, regWrite=1, rsValue=17, jump=0, memCLK=10, canJump=false, rtValue=18, zeroFlag=false, jumpAddress=-1, executeCLK=9, *ALUControl*=0010, oneFlag=true, *ALUresult*=306, memToReg=0, writebackCLK=11, PC=8, decodeCLK=8, memWrite=0, isImmediate=0, rdIndex=18, regDst=1, rtIndex=17, canBranch=false, branchAddress=0, opCode=0011}

Instruction Decode ... at ClockCycle: 9
ID_EX{*Instruction*=0100100111010010101000000000000, memRead=0, *JumpAddress*=1111001110100101010000000000000, fetchCLK=8, branch=0, immediate=1111111111101010000000000000000, regWrite=1, rsValue=20, jump=0, memCLK=11, rtValue=21, executeCLK=10, *ALUControl*=0100, memToReg=0, writebackCLK=12, PC=9, decodeCLK=9, memWrite=0, isImmediate=0, rdIndex=21, regDst=1, rtIndex=20, opCode=0100}

Instruction Fetching ... at Clock Cycle: 9
IF_ID{writebackCLK=13, memCLK=12, *Instruction*=100100010001110000000000000000100, fetchCLK=9, PC=10, decodeCLK=10, executeCLK=11}

ClockCycle: 10

WriteBack at ... ClockCycle: 10
MA_WB{*Instruction*=1000010010101000000000000000011, *memRead*=1, *JumpAddress*=0000100101010000000000000000011, *fetchCLK*=6, *branch*=0, *immediate*=00000000000000000000000000000011, *gWrite*=1, *rsValue*=10, *jump*=0, *memCLK*=9, *canJump*=false, *rtValue*=11, *zeroFlag*=false, *executeCLK*=8, *jumpAddress*=-1, *ALUControl*=0000, *memoryOutput*=14, *oneFlag*=true, *memToReg*=1, *ALUresult*=13, *writebackCLK*=10, *PC*=7, *decodeCLK*=7, *memWrite*=0, *isImmediate*=1, *rdIndex*=0, *regDst*=0, *rtIndex*=10, *opCode*=1000, *branchAddress*=0, *canBranch*=false}

Memory Access at ... ClockCycle: 10

No data to output from memory; e.g sw
MA_WB{*Instruction*=0011100010011001000000000000000, *memRead*=0, *JumpAddress*=0000100010011001000000000000000, *fetchCLK*=7, *branch*=0, *immediate*=0000000000001001000000000000000, *regWrite*=1, *rsValue*=17, *jump*=0, *memCLK*=10, *canJump*=false, *rtValue*=18, *zeroFlag*=false, *executeCLK*=9, *jumpAddress*=-1, *ALUControl*=0010, *memoryOutput*=-1, *oneFlag*=true, *memToReg*=0, *ALUresult*=306, *writebackCLK*=11, *PC*=8, *decodeCLK*=8, *memWrite*=0, *isImmediate*=0, *rdIndex*=18, *regDst*=1, *rtIndex*=17, *opCode*=0011, *branchAddress*=0, *canBranch*=false}

Execute at ... ClockCycle: 10

EX_MA{*Instruction*=0100100111010010101000000000000, *JumpAddress*=1111100111010010101000000000000, *memRead*=0, *fetchCLK*=8, *immediate*=1111111111111101010000000000000, *branch*=0, *regWrite*=1, *rsValue*=20, *jump*=0, *memCLK*=11, *canJump*=false, *rtValue*=21, *zeroFlag*=false, *executeCLK*=10, *ALUControl*=0100, *oneFlag*=true, *ALUresult*=20, *memToReg*=0, *writebackCLK*=12, *PC*=9, *decodeCLK*=9, *memWrite*=0, *isImmediate*=0, *rdIndex*=21, *regDst*=1, *rtIndex*=20, *canBranch*=false, *branchAddress*=0, *opCode*=0100}

Instruction Decode ... at ClockCycle: 10

ID_EX{*Instruction*=1001000100011100000000000000100, *memRead*=0, *JumpAddress*=00000001000111000000000000000100, *fetchCLK*=9, *branch*=0, *immediate*=000000000000000000000000000000100, *egWrite*=0, *rsValue*=3, *jump*=0, *memCLK*=12, *rtValue*=8, *executeCLK*=11, *ALUControl*=0000, *memToReg*=0, *writebackCLK*=13, *PC*=10, *decodeCLK*=10, *memWrite*=1, *isImmediate*=1, *rdIndex*=0, *regDst*=0, *rtIndex*=7, *opCode*=1001}

Instruction Fetching ... at Clock Cycle: 10

IF_ID{*writebackCLK*=14, *memCLK*=13, *Instruction*=0110101101011100000000000000000, *fetchCLK*=10, *PC*=11, *decodeCLK*=11, *executeCLK*=12}

ClockCycle: 11

WriteBack at ... ClockCycle: 11

MA_WB{*Instruction*=0011100010011001000000000000000, *memRead*=0, *JumpAddress*=0000100010011001000000000000000, *fetchCLK*=7, *branch*=0, *immediate*=0000000000001001000000000000000, *regWrite*=1, *rsValue*=17, *jump*=0, *memCLK*=10, *canJump*=false, *rtValue*=18, *zeroFlag*=false, *executeCLK*=9, *jumpAddress*=-1, *ALUControl*=0010, *memoryOutput*=-1, *oneFlag*=true, *memToReg*=0, *ALUresult*=306, *writebackCLK*=11, *PC*=8, *decodeCLK*=8, *memWrite*=0, *isImmediate*=0, *rdIndex*=18, *regDst*=1, *rtIndex*=17, *opCode*=0011, *branchAddress*=0, *canBranch*=false}

Memory Access at ... ClockCycle: 11

No data to output from memory; e.g sw
MA_WB{*Instruction*=0100100111010010101000000000000, *memRead*=0, *JumpAddress*=1111100111010010101000000000000, *fetchCLK*=8, *branch*=0, *immediate*=1111111111111101010000000000000, *regWrite*=1, *rsValue*=20, *jump*=0, *memCLK*=11, *canJump*=false, *rtValue*=21, *zeroFlag*=false, *executeCLK*=10, *jumpAddress*=-1, *ALUControl*=0100, *memoryOutput*=-1, *oneFlag*=true, *memToReg*=0, *ALUresult*=20, *writebackCLK*=12, *PC*=9, *decodeCLK*=9, *memWrite*=0, *isImmediate*=0, *rdIndex*=21, *regDst*=1, *rtIndex*=20, *opCode*=0100, *branchAddress*=0, *canBranch*=false}

Execute at ... ClockCycle: 11

EX_MA{*Instruction*=1001000100011100000000000000100, *JumpAddress*=0000000100011100000000000000000, *memRead*=0, *fetchCLK*=9, *immediate*=0000000000000000000000000000000, *branch*=0, *regWrite*=0, *rsValue*=3, *jump*=0, *memCLK*=12, *canJump*=false, *rtValue*=8, *zeroFlag*=false, *executeCLK*=11, *ALUControl*=0000, *oneFlag*=true, *ALUresult*=7, *memToReg*=0, *writebackCLK*=13, *PC*=10, *decodeCLK*=10, *memWrite*=1, *isImmediate*=1, *rdIndex*=0, *regDst*=0, *rtIndex*=7, *canBranch*=false, *branchAddress*=0, *opCode*=1001}

Instruction Decode ... at ClockCycle: 11

ID_EX{*Instruction*=0110101101011100000000000000000, *memRead*=0, *JumpAddress*=0000101101011100000000000000000, *fetchCLK*=10, *branch*=0, *immediate*=0000000000000000000000000000000, *egWrite*=1, *rsValue*=23, *jump*=0, *memCLK*=13, *rtValue*=24, *executeCLK*=12, *ALUControl*=0101, *memToReg*=0, *writebackCLK*=14, *PC*=11, *decodeCLK*=11, *memWrite*=0, *isImmediate*=1, *rdIndex*=0, *regDst*=0, *rtIndex*=23, *opCode*=0110}

Instruction Fetching ... at Clock Cycle: 11

IF_ID{*writebackCLK*=15, *memCLK*=14, *Instruction*=0111100011001000000000000000000, *fetchCLK*=11, *PC*=12, *decodeCLK*=12, *executeCLK*=13}

ClockCycle: 14

WriteBack at ... ClockCycle: 14

```
MA_WB{Instruction=01101011010111000000000000000010, memRead=0, JumpAddress=00001011010111000000000000000010, fetchCLK=10, branch=0, immediate=00000000000000000000000000000010,  
regWrite=1, rsValue=23, jump=0, memCLK=13, canJump=false, rtValue=24, zeroFlag=false, executeCLK=12, jumpAddress=-1, ALUControl=0101, memoryOutput=-1, oneFlag=true,  
memToReg=0, ALUresult=92, writebackCLK=14, PC=11, decodeCLK=11, memWrite=0, isImmediate=1, rdIndex=0, regDst=0, rtIndex=23, opCode=0110, branchAddress=0, canBranch=false}
```

Memory Access at ... ClockCycle: 14

No data to output from memory; e.g sw

```
MA_WB{Instruction=01111000110010000000000000000010, memRead=0, JumpAddress=00001100011001000000000000000010, fetchCLK=11, branch=0, immediate=00000000000000000000000000000010,  
regWrite=1, rsValue=25, jump=0, memCLK=14, canJump=false, rtValue=26, zeroFlag=false, executeCLK=13, jumpAddress=-1, ALUControl=0110, memoryOutput=-1, oneFlag=true,  
memToReg=0, ALUresult=6, writebackCLK=15, PC=12, decodeCLK=12, memWrite=0, isImmediate=1, rdIndex=0, regDst=0, rtIndex=25, opCode=0111, branchAddress=0, canBranch=false}
```

Execute at ... ClockCycle: 14

```
EX_MA{Instruction=100100010011100000000000000000111, JumpAddress=000000100111000000000000000000111, memRead=0, fetchCLK=12, immediate=00000000000000000000000000000000111,  
branch=0, regWrite=0, rsValue=3, jump=0, memCLK=15, canJump=false, rtValue=1, zeroFlag=false, executeCLK=14, ALUControl=0000, oneFlag=true,  
ALUresult=10, memToReg=0, writebackCLK=16, PC=13, decodeCLK=13, memWrite=1, isImmediate=1, rdIndex=0, regDst=0, rtIndex=15, canBranch=false, branchAddress=0, opCode=1001}
```

Instruction Decode ... at ClockCycle: 14

```
ID_EX{Instruction=10111000011100000000000000001111, memRead=0, JumpAddress=00001000011100000000000000001111, fetchCLK=13, branch=1, immediate=000000000000000000000000000000001111,  
regWrite=0, rsValue=17, jump=0, memCLK=16, rtValue=1, executeCLK=15, ALUControl=1000, memToReg=0, writebackCLK=17, PC=14, decodeCLK=14, memWrite=0, isImmediate=0,  
rdIndex=0, regDst=0, rtIndex=15, opCode=1011}
```

Instruction Fetching ... at Clock Cycle: 14

```
IF_ID{writebackCLK=18, memCLK=17, Instruction=1100111011011110000000000000, fetchCLK=14, PC=15, decodeCLK=15, executeCLK=16}
```

ClockCycle: 15

WriteBack at ... ClockCycle: 15

```
MA_WB{Instruction=01111000110010000000000000000010, memRead=0, JumpAddress=00001100011001000000000000000010, fetchCLK=11, branch=0, immediate=00000000000000000000000000000010,  
regWrite=1, rsValue=25, jump=0, memCLK=14, canJump=false, rtValue=26, zeroFlag=false, executeCLK=13, jumpAddress=-1, ALUControl=0110, memoryOutput=-1, oneFlag=true,  
memToReg=0, ALUresult=6, writebackCLK=15, PC=12, decodeCLK=12, memWrite=0, isImmediate=1, rdIndex=0, regDst=0, rtIndex=25, opCode=0111, branchAddress=0, canBranch=false}
```

Memory Access at ... ClockCycle: 15

data: 10

```
MA_WB{Instruction=100100010011100000000000000000111, memRead=0, JumpAddress=000000100111000000000000000000111, fetchCLK=12, branch=0, immediate=00000000000000000000000000000000111,  
regWrite=0, rsValue=3, jump=0, memCLK=15, canJump=false, rtValue=1, zeroFlag=false, executeCLK=14, jumpAddress=-1, ALUControl=0000, memoryOutput=-1, oneFlag=true,  
memToReg=0, ALUresult=10, writebackCLK=16, PC=13, decodeCLK=13, memWrite=1, isImmediate=1, rdIndex=0, regDst=0, rtIndex=15, opCode=1001, branchAddress=0, canBranch=false}
```

Execute at ... ClockCycle: 15

```
EX_MA{Instruction=10111000011100000000000000001111, JumpAddress=00001000011100000000000000001111, memRead=0, fetchCLK=13, immediate=000000000000000000000000000000001111,  
branch=1, regWrite=0, rsValue=17, jump=0, memCLK=16, canJump=false, rtValue=1, zeroFlag=false, executeCLK=15, ALUControl=1000, oneFlag=true,  
ALUresult=1, memToReg=0, writebackCLK=17, PC=14, decodeCLK=14, memWrite=0, isImmediate=0, rdIndex=0, regDst=0, rtIndex=15, canBranch=true, branchAddress=15, opCode=1011}
```

Instruction Decode ... at ClockCycle: 15

```
ID_EX{Instruction=1100111011011110000000000000, memRead=0, JumpAddress=11111110110111100000000000, fetchCLK=14, branch=0, immediate=11111111111111000000000000,  
regWrite=1, rsValue=31, jump=0, memCLK=17, rtValue=30, executeCLK=16, ALUControl=0111, memToReg=0, writebackCLK=18, PC=15, decodeCLK=15, memWrite=0, isImmediate=0,  
rdIndex=31, regDst=1, rtIndex=29, opCode=1100}
```

Instruction Fetching ... at Clock Cycle: 15

```
IF_ID{writebackCLK=19, memCLK=18, Instruction=1100110101101111000000000000, fetchCLK=15, PC=16, decodeCLK=16, executeCLK=17}
```

ClockCycle: 16

WriteBack at ... *ClockCycle:* 16

There was no data to write back

MA_WB{*Instruction*=100100010011110000000000000000111, *memRead*=0, *JumpAddress*=000000010011110000000000000000111, *fetchCLK*=12, *branch*=0, *immediate*=000000000000000000000000000000111, *regWrite*=0, *rsValue*=3, *jump*=0, *memCLK*=15, *canJump*=false, *rtValue*=1, *zeroFlag*=false, *executeCLK*=14, *jumpAddress*=-1, *ALUControl*=0000, *memoryOutput*=-1, *oneFlag*=true, *memToReg*=0, *ALUresult*=10, *writebackCLK*=16, *PC*=13, *decodeCLK*=13, *memWrite*=1, *isImmediate*=1, *rdIndex*=0, *regDst*=0, *rtIndex*=15, *opCode*=1001, *branchAddress*=0, *canBranch*=false}

Memory Access at ... *ClockCycle:* 16

No data to output from memory; e.g sw

MA_WB{*Instruction*=101110000011110000000000000000111, *memRead*=0, *JumpAddress*=000010000011110000000000000000111, *fetchCLK*=13, *branch*=1, *immediate*=000000000000000000000000000000111, *regWrite*=0, *rsValue*=17, *jump*=0, *memCLK*=16, *canJump*=false, *rtValue*=1, *zeroFlag*=false, *executeCLK*=15, *jumpAddress*=-1, *ALUControl*=1000, *memoryOutput*=-1, *oneFlag*=true, *memToReg*=0, *ALUresult*=1, *writebackCLK*=17, *PC*=14, *decodeCLK*=14, *memWrite*=0, *isImmediate*=0, *rdIndex*=0, *regDst*=0, *rtIndex*=15, *opCode*=1011, *branchAddress*=15, *canBranch*=true}

EX_MA{}

ID_EX{}

Instruction Fetching ... at *Clock Cycle:* 16

IF_ID{*writebackCLK*=20, *memCLK*=19, *Instruction*=11001101011011110000000000000000, *fetchCLK*=16, *PC*=16, *decodeCLK*=17, *executeCLK*=18}

ClockCycle: 17

WriteBack at ... *ClockCycle:* 17

There was no data to write back

MA_WB{*Instruction*=101110000011110000000000000000111, *memRead*=0, *JumpAddress*=000010000011110000000000000000111, *fetchCLK*=13, *branch*=1, *immediate*=000000000000000000000000000000111, *regWrite*=0, *rsValue*=17, *jump*=0, *memCLK*=16, *canJump*=false, *rtValue*=1, *zeroFlag*=false, *executeCLK*=15, *jumpAddress*=-1, *ALUControl*=1000, *memoryOutput*=-1, *oneFlag*=true, *memToReg*=0, *ALUresult*=1, *writebackCLK*=17, *PC*=14, *decodeCLK*=14, *memWrite*=0, *isImmediate*=0, *rdIndex*=0, *regDst*=0, *rtIndex*=15, *opCode*=1011, *branchAddress*=15, *canBranch*=true}

MA_WB{*Instruction*=101110000011110000000000000000111, *memRead*=0, *JumpAddress*=000010000011110000000000000000111, *fetchCLK*=13, *branch*=1, *immediate*=000000000000000000000000000000111, *regWrite*=0, *rsValue*=17, *jump*=0, *memCLK*=16, *canJump*=false, *rtValue*=1, *zeroFlag*=false, *executeCLK*=15, *jumpAddress*=-1, *ALUControl*=1000, *memoryOutput*=-1, *oneFlag*=true, *memToReg*=0, *ALUresult*=1, *writebackCLK*=17, *PC*=14, *decodeCLK*=14, *memWrite*=0, *isImmediate*=0, *rdIndex*=0, *regDst*=0, *rtIndex*=15, *opCode*=1011, *branchAddress*=15, *canBranch*=true}

EX_MA{}

Instruction Decode ... at *ClockCycle:* 17

ID_EX{*Instruction*=11001101011011110000000000000000, *memRead*=0, *JumpAddress*=111110101101111000000000000000, *fetchCLK*=16, *branch*=0, *immediate*=1111111111111100000000000000, *regWrite*=1, *rsValue*=27, *jump*=0, *memCLK*=19, *rtValue*=28, *executeCLK*=18, *ALUControl*=0111, *memToReg*=0, *writebackCLK*=20, *PC*=16, *decodeCLK*=17, *memWrite*=0, *isImmediate*=0, *rdIndex*=28, *regDst*=1, *rtIndex*=27, *opCode*=1100}

Instruction Fetching ... at *Clock Cycle:* 17

IF_ID{*writebackCLK*=20, *memCLK*=19, *Instruction*=11001101011011110000000000000000, *fetchCLK*=16, *PC*=16, *decodeCLK*=17, *executeCLK*=18}

ClockCycle: 18

WriteBack at ... ClockCycle: 18
There was no data to write back
MA_WB{*Instruction*=101110000011110000000000000000001111, *memRead*=0, *JumpAddress*=000010000011110000000000000000001111, *fetchCLK*=13, *branch*=1, *immediate*=000000000000000000000000000000001111, *regWrite*=0, *rsValue*=17, *jump*=0, *memCLK*=16, *canJump*=false, *rtValue*=1, *zeroFlag*=false, *executeCLK*=15, *jumpAddress*=-1, *ALUControl*=1000, *memoryOutput*=-1, *oneFlag*=true, *memToReg*=0, *ALUresult*=1, *writebackCLK*=17, *PC*=14, *decodeCLK*=14, *memWrite*=0, *isImmediate*=0, *rdIndex*=0, *regDst*=0, *rtIndex*=15, *opCode*=1011, *branchAddress*=15, *canBranch*=true}

MA_WB{*Instruction*=101110000011110000000000000000001111, *memRead*=0, *JumpAddress*=000010000011110000000000000000001111, *fetchCLK*=13, *branch*=1, *immediate*=000000000000000000000000000000001111, *regWrite*=0, *rsValue*=17, *jump*=0, *memCLK*=16, *canJump*=false, *rtValue*=1, *zeroFlag*=false, *executeCLK*=15, *jumpAddress*=-1, *ALUControl*=1000, *memoryOutput*=-1, *oneFlag*=true, *memToReg*=0, *ALUresult*=1, *writebackCLK*=17, *PC*=14, *decodeCLK*=14, *memWrite*=0, *isImmediate*=0, *rdIndex*=0, *regDst*=0, *rtIndex*=15, *opCode*=1011, *branchAddress*=15, *canBranch*=true}

Execute at ... ClockCycle: 18
EX_MA{*Instruction*=11001101011011110000000000000000, *JumpAddress*=11111010110111100000000000000000, *memRead*=0, *fetchCLK*=16, *immediate*=11111111111111110000000000000000, *branch*=0, *regWrite*=1, *rsValue*=27, *jump*=0, *memCLK*=19, *canJump*=false, *rtValue*=28, *zeroFlag*=false, *jumpAddress*=-1, *executeCLK*=18, *ALUControl*=0111, *oneFlag*=true, *ALUresult*=1, *memToReg*=0, *writebackCLK*=20, *PC*=16, *decodeCLK*=17, *memWrite*=0, *isImmediate*=0, *rdIndex*=28, *regDst*=1, *rtIndex*=27, *canBranch*=false, *branchAddress*=0, *opCode*=1100}

ID_EX{*Instruction*=11001101011011110000000000000000, *memRead*=0, *JumpAddress*=11111010110111100000000000000000, *fetchCLK*=16, *branch*=0, *immediate*=11111111111111110000000000000000, *regWrite*=1, *rsValue*=27, *jump*=0, *memCLK*=19, *rtValue*=28, *executeCLK*=18, *ALUControl*=0111, *memToReg*=0, *writebackCLK*=20, *PC*=16, *decodeCLK*=17, *memWrite*=0, *isImmediate*=0, *rdIndex*=28, *regDst*=1, *rtIndex*=27, *opCode*=1100}

Instruction Fetching ... at Clock Cycle: 18

IF_ID{*writebackCLK*=20, *memCLK*=19, *Instruction*=11001101011011110000000000000000, *fetchCLK*=16, *PC*=16, *decodeCLK*=17, *executeCLK*=18}

ClockCycle: 19

WriteBack at ... ClockCycle: 19
There was no data to write back
MA_WB{*Instruction*=101110000011110000000000000000001111, *memRead*=0, *JumpAddress*=000010000011110000000000000000001111, *fetchCLK*=13, *branch*=1, *immediate*=000000000000000000000000000000001111, *regWrite*=0, *rsValue*=17, *jump*=0, *memCLK*=16, *canJump*=false, *rtValue*=1, *zeroFlag*=false, *executeCLK*=15, *jumpAddress*=-1, *ALUControl*=1000, *memoryOutput*=-1, *oneFlag*=true, *memToReg*=0, *ALUresult*=1, *writebackCLK*=17, *PC*=14, *decodeCLK*=14, *memWrite*=0, *isImmediate*=0, *rdIndex*=0, *regDst*=0, *rtIndex*=15, *opCode*=1011, *branchAddress*=15, *canBranch*=true}

Memory Access at ... ClockCycle: 19
No data to output from memory; e.g sw
MA_WB{*Instruction*=11001101011011110000000000000000, *memRead*=0, *JumpAddress*=11111010110111100000000000000000, *fetchCLK*=16, *branch*=0, *immediate*=11111111111111110000000000000000, *regWrite*=1, *rsValue*=27, *jump*=0, *memCLK*=19, *canJump*=false, *rtValue*=28, *zeroFlag*=false, *executeCLK*=18, *jumpAddress*=-1, *ALUControl*=0111, *memoryOutput*=-1, *oneFlag*=true, *memToReg*=0, *ALUresult*=1, *writebackCLK*=20, *PC*=16, *decodeCLK*=17, *memWrite*=0, *isImmediate*=0, *rdIndex*=28, *regDst*=1, *rtIndex*=27, *opCode*=1100, *branchAddress*=0, *canBranch*=false}

EX_MA{*Instruction*=11001101011011110000000000000000, *JumpAddress*=11111010110111100000000000000000, *memRead*=0, *fetchCLK*=16, *immediate*=11111111111111110000000000000000, *branch*=0, *regWrite*=1, *rsValue*=27, *jump*=0, *memCLK*=19, *canJump*=false, *rtValue*=28, *zeroFlag*=false, *executeCLK*=18, *ALUControl*=0111, *oneFlag*=true, *ALUresult*=-1, *memToReg*=0, *writebackCLK*=20, *PC*=16, *decodeCLK*=17, *memWrite*=0, *isImmediate*=0, *rdIndex*=28, *regDst*=1, *rtIndex*=27, *canBranch*=false, *branchAddress*=0, *opCode*=1100}

ID_EX{*Instruction*=11001101011011110000000000000000, *memRead*=0, *JumpAddress*=11111010110111100000000000000000, *fetchCLK*=16, *branch*=0, *immediate*=11111111111111110000000000000000, *regWrite*=1, *rsValue*=27, *jump*=0, *memCLK*=19, *rtValue*=28, *executeCLK*=18, *ALUControl*=0111, *memToReg*=0, *writebackCLK*=20, *PC*=16, *decodeCLK*=17, *memWrite*=0, *isImmediate*=0, *rdIndex*=28, *regDst*=1, *rtIndex*=27, *opCode*=1100}

Instruction Fetching ... at Clock Cycle: 19

IF_ID{*writebackCLK*=20, *memCLK*=19, *Instruction*=11001101011011110000000000000000, *fetchCLK*=16, *PC*=16, *decodeCLK*=17, *executeCLK*=18}

ClockCycle: 20

WriteBack at ... ClockCycle: 20

```
MA_WB{Instruction=11001101011011110000000000000000, memRead=0, JumpAddress=11111010110111100000000000000000, fetchCLK=16, branch=0, immediate=11111111111111000000000000000000, regWrite=1, rsValue=27, jump=0, memCLK=19, canJump=false, rtValue=28, zeroFlag=false, executeCLK=18, jumpAddress=-1, ALUControl=0111, memoryOutput=-1, oneFlag=true, memToReg=0, ALUresult=1, writebackCLK=20, PC=16, decodeCLK=17, memWrite=0, isImmediate=0, rdIndex=28, regDst=1, rtIndex=27, opCode=1100, branchAddress=0, canBranch=false}
```

```
MA_WB{Instruction=11001101011011110000000000000000, memRead=0, JumpAddress=11111010110111100000000000000000, fetchCLK=16, branch=0, immediate=11111111111111000000000000000000, regWrite=1, rsValue=27, jump=0, memCLK=19, canJump=false, rtValue=28, zeroFlag=false, executeCLK=18, jumpAddress=-1, ALUControl=0111, memoryOutput=-1, oneFlag=true, memToReg=0, ALUresult=1, writebackCLK=20, PC=16, decodeCLK=17, memWrite=0, isImmediate=0, rdIndex=28, regDst=1, rtIndex=27, opCode=1100, branchAddress=0, canBranch=false}
```

```
EX_MA{Instruction=11001101011011110000000000000000, JumpAddress=11111010110111100000000000000000, memRead=0, fetchCLK=16, immediate=11111111111111000000000000000000, branch=0, regWrite=1, rsValue=27, jump=0, memCLK=19, canJump=false, rtValue=28, zeroFlag=false, jumpAddress=-1, executeCLK=18, ALUControl=0111, oneFlag=true, ALUresult=-1, memToReg=0, writebackCLK=20, PC=16, decodeCLK=17, memWrite=0, isImmediate=0, rdIndex=28, regDst=1, rtIndex=27, canBranch=false, branchAddress=0, opCode=1100}
```

```
ID_EX{Instruction=11001101011011110000000000000000, memRead=0, JumpAddress=11111010110111100000000000000000, fetchCLK=16, branch=0, immediate=11111111111111000000000000000000, regWrite=1, rsValue=27, jump=0, memCLK=19, rtValue=28, executeCLK=18, ALUControl=0111, memToReg=0, writebackCLK=20, PC=16, decodeCLK=17, memWrite=0, isImmediate=0, rdIndex=28, regDst=1, rtIndex=27, opCode=1100}
```

Instruction Fetching ... at Clock Cycle: 20

```
IF_ID{writebackCLK=20, memCLK=19, Instruction=11001101011011110000000000000000, fetchCLK=16, PC=16, decodeCLK=17, executeCLK=18}
```

ClockCycle: 21

```
MA_WB{Instruction=11001101011011110000000000000000, memRead=0, JumpAddress=11111010110111100000000000000000, fetchCLK=16, branch=0, immediate=11111111111111000000000000000000, regWrite=1, rsValue=27, jump=0, memCLK=19, canJump=false, rtValue=28, zeroFlag=false, executeCLK=18, jumpAddress=-1, ALUControl=0111, memoryOutput=-1, oneFlag=true, memToReg=0, ALUresult=1, writebackCLK=20, PC=16, decodeCLK=17, memWrite=0, isImmediate=0, rdIndex=28, regDst=1, rtIndex=27, opCode=1100, branchAddress=0, canBranch=false}
```

```
MA_WB{Instruction=11001101011011110000000000000000, memRead=0, JumpAddress=11111010110111100000000000000000, fetchCLK=16, branch=0, immediate=11111111111111000000000000000000, regWrite=1, rsValue=27, jump=0, memCLK=19, canJump=false, rtValue=28, zeroFlag=false, executeCLK=18, jumpAddress=-1, ALUControl=0111, memoryOutput=-1, oneFlag=true, memToReg=0, ALUresult=1, writebackCLK=20, PC=16, decodeCLK=17, memWrite=0, isImmediate=0, rdIndex=28, regDst=1, rtIndex=27, opCode=1100, branchAddress=0, canBranch=false}
```

```
EX_MA{Instruction=11001101011011110000000000000000, JumpAddress=11111010110111100000000000000000, memRead=0, fetchCLK=16, immediate=11111111111111000000000000000000, branch=0, regWrite=1, rsValue=27, jump=0, memCLK=19, canJump=false, rtValue=28, zeroFlag=false, jumpAddress=-1, executeCLK=18, ALUControl=0111, oneFlag=true, ALUresult=-1, memToReg=0, writebackCLK=20, PC=16, decodeCLK=17, memWrite=0, isImmediate=0, rdIndex=28, regDst=1, rtIndex=27, canBranch=false, branchAddress=0, opCode=1100}
```

```
ID_EX{Instruction=11001101011011110000000000000000, memRead=0, JumpAddress=11111010110111100000000000000000, fetchCLK=16, branch=0, immediate=11111111111111000000000000000000, regWrite=1, rsValue=27, jump=0, memCLK=19, rtValue=28, executeCLK=18, ALUControl=0111, memToReg=0, writebackCLK=20, PC=16, decodeCLK=17, memWrite=0, isImmediate=0, rdIndex=28, regDst=1, rtIndex=27, opCode=1100}
```

Instruction Fetching ... at Clock Cycle: 21

```
IF_ID{writebackCLK=20, memCLK=19, Instruction=11001101011011110000000000000000, fetchCLK=16, PC=16, decodeCLK=17, executeCLK=18}
```

```
Data memory 0 : 1
Data memory 1 : 10
Data memory 2 : 3
Data memory 3 : 4
Data memory 4 : 5
Data memory 5 : 6
Data memory 6 : 7
Data memory 7 : 8
Data memory 8 : 7
Data memory 9 : 10
Data memory 10 : 11
Data memory 11 : 12
Data memory 12 : 13
Data memory 13 : 14
Data memory 14 : 15
Data memory 15 : 16
Data memory 16 : 17
Data memory 17 : 18
Data memory 18 : 19
Data memory 19 : 20
```

```
Register File 0 : NAME: reg0, DESCRIPTION: General Purpose, VALUE: 1
Register File 1 : NAME: reg1, DESCRIPTION: General Purpose, VALUE: 2
Register File 2 : NAME: reg2, DESCRIPTION: General Purpose, VALUE: 3
Register File 3 : NAME: reg3, DESCRIPTION: General Purpose, VALUE: 4
Register File 4 : NAME: reg4, DESCRIPTION: General Purpose, VALUE: 5
Register File 5 : NAME: reg5, DESCRIPTION: General Purpose, VALUE: 6
Register File 6 : NAME: reg6, DESCRIPTION: General Purpose, VALUE: 7
Register File 7 : NAME: reg7, DESCRIPTION: General Purpose, VALUE: 8
Register File 8 : NAME: reg8, DESCRIPTION: General Purpose, VALUE: 11
Register File 9 : NAME: reg9, DESCRIPTION: General Purpose, VALUE: 10
Register File 10 : NAME: reg10, DESCRIPTION: General Purpose, VALUE: 14
Register File 11 : NAME: reg11, DESCRIPTION: General Purpose, VALUE: 12
Register File 12 : NAME: reg12, DESCRIPTION: General Purpose, VALUE: 10
Register File 13 : NAME: reg13, DESCRIPTION: General Purpose, VALUE: 14
Register File 14 : NAME: reg14, DESCRIPTION: General Purpose, VALUE: 15
Register File 15 : NAME: reg15, DESCRIPTION: General Purpose, VALUE: 1
Register File 16 : NAME: reg16, DESCRIPTION: General Purpose, VALUE: 17
Register File 17 : NAME: reg17, DESCRIPTION: General Purpose, VALUE: 18
Register File 18 : NAME: reg18, DESCRIPTION: General Purpose, VALUE: 306
Register File 19 : NAME: reg19, DESCRIPTION: General Purpose, VALUE: 20
Register File 20 : NAME: reg20, DESCRIPTION: General Purpose, VALUE: 21
Register File 21 : NAME: reg21, DESCRIPTION: General Purpose, VALUE: 20
Register File 22 : NAME: reg22, DESCRIPTION: General Purpose, VALUE: 23
Register File 23 : NAME: reg23, DESCRIPTION: General Purpose, VALUE: 92
Register File 24 : NAME: reg24, DESCRIPTION: General Purpose, VALUE: 25
Register File 25 : NAME: reg25, DESCRIPTION: General Purpose, VALUE: 6
Register File 26 : NAME: reg26, DESCRIPTION: General Purpose, VALUE: 27
Register File 27 : NAME: reg27, DESCRIPTION: General Purpose, VALUE: 28
Register File 28 : NAME: reg28, DESCRIPTION: General Purpose, VALUE: 1
Register File 29 : NAME: reg29, DESCRIPTION: General Purpose, VALUE: 30
Register File 30 : NAME: reg30, DESCRIPTION: General Purpose, VALUE: 31
Register File 31 : NAME: reg31, DESCRIPTION: General Purpose, VALUE: 32
```

```
Cache 0 :Valid: true, Tag: 1, Data: 7
Cache 1 :Valid: true, Tag: 0, Data: 10
Cache 2 :Valid: false, Tag: -1, Data: -1
Cache 3 :Valid: false, Tag: -1, Data: -1
Cache 4 :Valid: false, Tag: -1, Data: -1
Cache 5 :Valid: true, Tag: 1, Data: 14
Cache 6 :Valid: false, Tag: -1, Data: -1
Cache 7 :Valid: false, Tag: -1, Data: -1
```