

Report

Ice_ahh_fakher_bygeeb_mel_a5er

Features:

1. Micro-architecture: Harvard
2. Size of instruction and data memory: 1024 x 32-bit
3. Total number of registers: 32 Registers
4. Instruction sets: Instruction set 1

Instruction format:

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

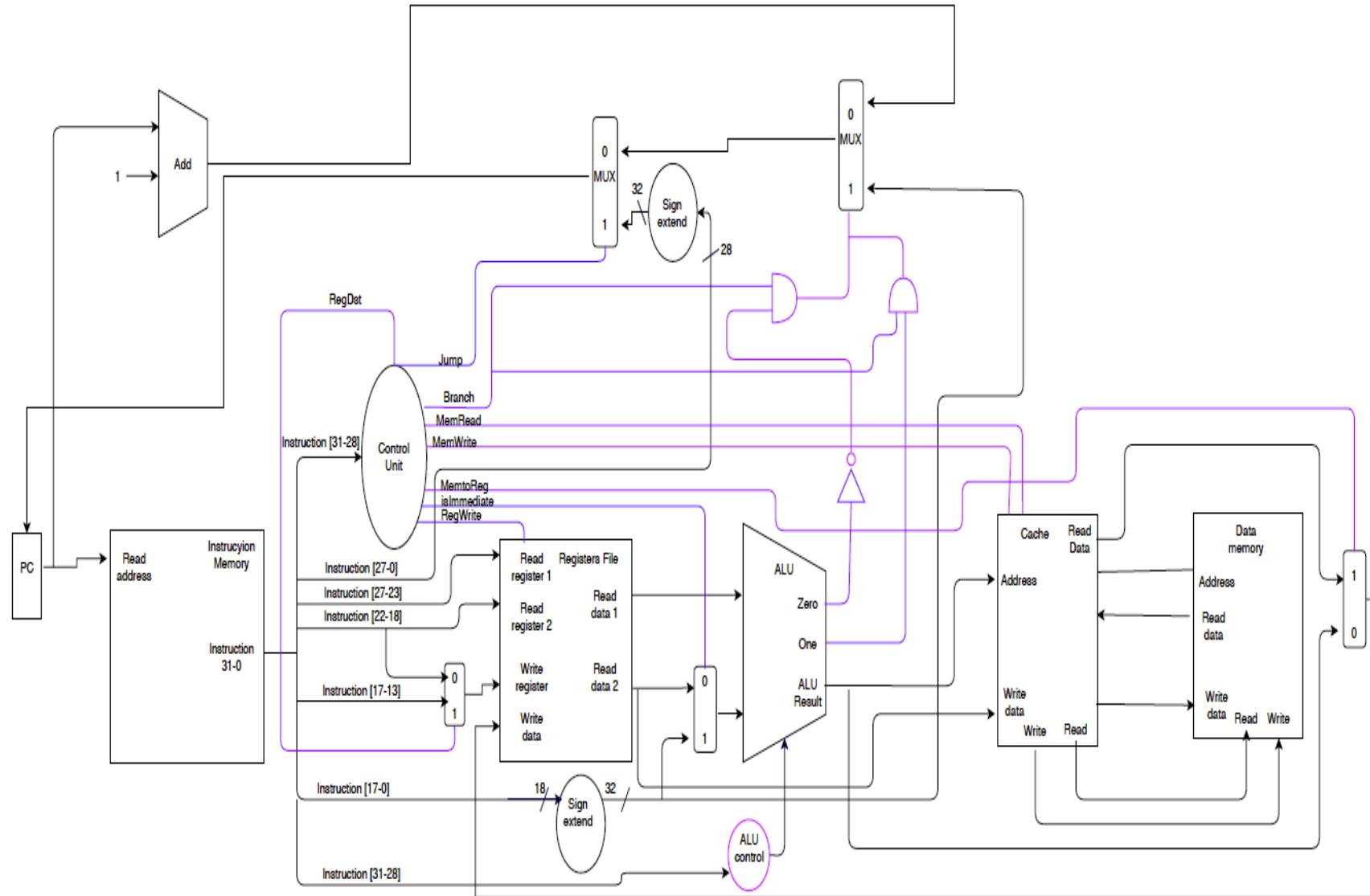
J-TYPE

<i>OPCODE</i>	JUMP-ADDRESS
31 -> 28	27 -> 0

cache type:

The cache we used is 8 places directly-mapped cache, which consequently makes the replacement policy to be the-oldest-one-out replacement policy, please notice the policy used in writing is “Write-through” policy, meaning that, when a write is made to the cache, the cache goes on and writes to the main memory, by this we avoid data inconsistencies between the cache and the main memory.

Data Path



notice you can find in the data path that the cache communicates with the data memory with two lines one called “write” and the other called “read” those lines are there as a way for the cache to control the data memory.

Code Structure

The project directory comes with two packages:

1. “components” package: here you can find **most** (not all) of the hardware-simulated components.
2. “stages” package: here you can find some of the hardware-simulated components as well as data-path stages (fetch, decode, execute, memory access, and write-back), and also a class called “Simulator” in that class you can find the main method which is the engine for running the code, in the main method we are basically creating a new simulator object which loads the program in the instruction memory our program has the following instructions :

Please ignore the comments

```
String bne = "10100000000010000000000000000011"; // bne reg0 reg1 imm=3      (goes to J)
String sw = "100100010001110000000000000000100"; // sw reg2 reg7 imm=4      (M[reg7]= reg2+4)
String add = "00000010000101001100000000000000"; // add reg4 reg5 reg6      (reg6=reg4+reg5)
String j = "110100000000000000000000000000101"; // jump imm = 5      (goes to LW)
String ori = "01010011101000000000000000000011"; // ori reg7 reg8 imm=3      (reg8=reg7 or 3)
String lw = "1000010010101000000000000000000011"; // lw reg9 reg10 imm=3      (reg10=M[reg9+3])
String addi ="000101011011001111111111111110"; // addi reg11 reg12 imm= -2    (reg12= reg11 - 2)
String sub = "00100111001101011110000000000000"; // sub reg14 reg13 reg15    (reg15=reg14-reg13)
String mult= "00111000010001100100000000000000"; // mult reg16 reg 17 reg18   (reg18=reg16*reg17)
String and = "01001001110100101010000000000000"; // and reg19 reg20 reg21    (reg21=reg19 and reg20 (result 20))
String sll = "01101011010111000000000000000010"; // sll reg22 reg23 imm=2     (reg23 = reg22 << 2) result = 92
String srl = "01111100011001000000000000000010"; // srl reg24 reg25 imm=2     (reg25 = reg24 >> 2) result = 6
String slt = "11001101011011110000000000000000"; // slt reg26 reg27 reg28    (reg28 = boolean reg26 < reg27) , true
String slt2= "11001111011101111100000000000000"; // slt reg30 reg29 reg31    (reg31 = boolean reg30 < reg29) , false
String sbt = "101110000111100000000000000000111"; // sbt reg16 reg15 imm=15   (skips slt2)
String sw2 = "100100010011100000000000000000111"; // sw reg2 reg15 imm=7      (M[reg15]= reg2+7)
```

Type to enter a caption.

Output

```

ClockCycle: 4

WriteBack at ... ClockCycle: 4
There was no data to write back
MA_WB{Instruction=10100000000010000000000000000001, JumpAddress=00000000000010000000000000000001, memRead=0, fetchCLK=0, branch=1, immediate=0000000000000000000000000000000011, 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}

MA_WB{Instruction=101000000000100000000000000000011, JumpAddress=000000000000100000000000000000011, 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}
EX_MA{ }

Instruction Decode ... at ClockCycle: 4
ID_EX{Instruction=010100111010000000000000000000011, memRead=0, JumpAddress=000001110100000000000000000000011, fetchCLK=3, branch=0, immediate=0000000000000000000000000000000011, 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=000101011011001111111111111110, fetchCLK=4, PC=5, decodeCLK=5, executeCLK=6}

ClockCycle: 5

WriteBack at ... ClockCycle: 5
There was no data to write back
MA_WB{Instruction=101000000000100000000000000000011, JumpAddress=000000000000100000000000000000011, memRead=0, fetchCLK=0, branch=1, immediate=0000000000000000000000000000000011, gWrite=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}

MA_WB{Instruction=101000000000100000000000000000011, JumpAddress=000000000000100000000000000000011, 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}

Execute at ... ClockCycle: 5
EX_MA{Instruction=010100111010000000000000000000011, JumpAddress=000001110100000000000000000000011, memRead=0, fetchCLK=3, immediate=0000000000000000000000000000000011, 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=000101011011001111111111111110, memRead=0, JumpAddress=111101011011001111111111111110, fetchCLK=4, branch=0, immediate=11111111111111111111111111111110, 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=001001110011011110000000000000, fetchCLK=5, PC=6, decodeCLK=6, executeCLK=7}

```

Type to enter a caption.

```

ClockCycle: 12

WriteBack at ... ClockCycle: 12
MA_WB{Instruction=01001001110100101010000000000000, memRead=0, JumpAddress=1111100111010010101000000000000, fetchCLK=8, branch=0, immediate=11111111111111010100000000000000, 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}

Memory Access at ... ClockCycle: 12
MA_WB{Instruction=10010001000111000000000000000000, memRead=0, JumpAddress=00000001000111000000000000000000, fetchCLK=9, branch=0, immediate=00000000000000000000000000000000, regWrite=0, rsValue=3, jump=0, memCLK=12, canJump=false, rtValue=8, zeroFlag=false, executeCLK=11, jumpAddress=-1, ALUControl=0000, memoryOutput=-1, oneFlag=true, memToReg=0, ALUresult=7, writebackCLK=13, PC=10, decodeCLK=10, memWrite=1, isImmediate=1, rdIndex=0, regDst=0, rtIndex=7, opCode=1001, branchAddress=0, canBranch=false}

Execute at ... ClockCycle: 12
EX_MA{Instruction=01101011010111000000000000000000, JumpAddress=00001011010111000000000000000000, memRead=0, fetchCLK=10, immediate=00000000000000000000000000000000, branch=0, regWrite=1, rsValue=23, jump=0, memCLK=13, canJump=false, rtValue=24, zeroFlag=false, jumpAddress=-1, executeCLK=12, ALUControl=0101, oneFlag=true, ALUresult=92, memToReg=0, writebackCLK=14, PC=11, decodeCLK=11, memWrite=0, isImmediate=1, rdIndex=0, regDst=0, rtIndex=23, canBranch=false, branchAddress=0, opCode=0110}data: 7

Instruction Decode ... at ClockCycle: 12
ID_EX{Instruction=01111100011001000000000000000000, memRead=0, JumpAddress=00001100011001000000000000000000, fetchCLK=11, branch=0, immediate=00000000000000000000000000000000, regWrite=1, rsValue=25, jump=0, memCLK=14, rtValue=26, executeCLK=13, ALUControl=0110, memToReg=0, writebackCLK=15, PC=12, decodeCLK=12, memWrite=0, isImmediate=1, rdIndex=0, regDst=0, rtIndex=25, opCode=0111}

Instruction Fetching ... at Clock Cycle: 12
IF_ID{writebackCLK=16, memCLK=15, Instruction=10010001001111000000000000000000111, fetchCLK=12, PC=13, decodeCLK=13, executeCLK=14}

ClockCycle: 13

WriteBack at ... ClockCycle: 13
There was no data to write back
MA_WB{Instruction=10010001000111000000000000000000, memRead=0, JumpAddress=00000001000111000000000000000000, fetchCLK=9, branch=0, immediate=00000000000000000000000000000000, regWrite=0, rsValue=3, jump=0, memCLK=12, canJump=false, rtValue=8, zeroFlag=false, executeCLK=11, jumpAddress=-1, ALUControl=0000, memoryOutput=-1, oneFlag=true, memToReg=0, ALUresult=7, writebackCLK=13, PC=10, decodeCLK=10, memWrite=1, isImmediate=1, rdIndex=0, regDst=0, rtIndex=7, opCode=1001, branchAddress=0, canBranch=false}

Memory Access at ... ClockCycle: 13
No data to output from memory; e.g. sw
MA_WB{Instruction=01101011010111000000000000000000, memRead=0, JumpAddress=00001011010111000000000000000000, fetchCLK=10, branch=0, immediate=00000000000000000000000000000000, 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}

Execute at ... ClockCycle: 13
EX_MA{Instruction=01111100011001000000000000000000, JumpAddress=00001100011001000000000000000000, memRead=0, fetchCLK=11, immediate=00000000000000000000000000000000, branch=0, regWrite=1, rsValue=25, jump=0, memCLK=14, canJump=false, rtValue=26, zeroFlag=false, jumpAddress=-1, executeCLK=13, ALUControl=0110, oneFlag=true, ALUresult=6, memToReg=0, writebackCLK=15, PC=12, decodeCLK=12, memWrite=0, isImmediate=1, rdIndex=0, regDst=0, rtIndex=25, canBranch=false, branchAddress=0, opCode=0111}

Instruction Decode ... at ClockCycle: 13
ID_EX{Instruction=10010001001111000000000000000000111, memRead=0, JumpAddress=00000001000111000000000000000000111, fetchCLK=12, branch=0, immediate=00000000000000000000000000000000, regWrite=0, rsValue=3, jump=0, memCLK=15, rtValue=1, executeCLK=14, ALUControl=0000, memToReg=0, writebackCLK=16, PC=13, decodeCLK=13, memWrite=1, isImmediate=1, rdIndex=0, regDst=0, rtIndex=15, opCode=1001}

Instruction Fetching ... at Clock Cycle: 13
IF_ID{writebackCLK=17, memCLK=16, Instruction=101110000111100000000000000000001111, fetchCLK=13, PC=14, decodeCLK=14, executeCLK=15}

```

Type to enter a caption.

Type to enter a caption

Type to enter a caption.

Type to enter a caption.

```

ClockCycle: 20

WriteBack at ... ClockCycle: 20
MA_WB{Instruction=11001101011011110000000000000000, memRead=0, JumpAddress=11111101011011110000000000000000, 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, memWwrite=0, isImmediate=0, rdIndex=28, regDst=1, rtIndex=27, opCode=1100, branchAddress=0, canBranch=false}

MA_WB{Instruction=11001101011011110000000000000000, memRead=0, JumpAddress=11111101011011110000000000000000, 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, memWwrite=0, isImmediate=0, rdIndex=28, regDst=1, rtIndex=27, opCode=1100, branchAddress=0, canBranch=false}

EX_MA{Instruction=11001101011011110000000000000000, JumpAddress=11111101011011110000000000000000, memRead=0, fetchCLK=16, immediate=11111111111111110000000000000000, branch=0, regWrite=1, rsValue=27, jump=0, memCLK=19, canJump=false, rtValue=28, zeroFlag=false, executeCLK=18, jumpAddress=-1, ALUControl=0111, oneFlag=true, ALUresult=-1, memToReg=0, writebackCLK=20, PC=16, decodeCLK=17, memWwrite=0, isImmediate=0, rdIndex=28, regDst=1, rtIndex=27, canBranch=false, branchAddress=0, opCode=1100}

ID_EX{Instruction=11001101011011110000000000000000, memRead=0, JumpAddress=11111101011011110000000000000000, 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, memWwrite=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=11111101011011110000000000000000, 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, memWwrite=0, isImmediate=0, rdIndex=28, regDst=1, rtIndex=27, opCode=1100, branchAddress=0, canBranch=false}

MA_WB{Instruction=11001101011011110000000000000000, memRead=0, JumpAddress=11111101011011110000000000000000, 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, memWwrite=0, isImmediate=0, rdIndex=28, regDst=1, rtIndex=27, opCode=1100, branchAddress=0, canBranch=false}

EX_MA{Instruction=11001101011011110000000000000000, JumpAddress=11111101011011110000000000000000, memRead=0, fetchCLK=16, immediate=11111111111111110000000000000000, branch=0, regWrite=1, rsValue=27, jump=0, memCLK=19, canJump=false, rtValue=28, zeroFlag=false, executeCLK=18, jumpAddress=-1, ALUControl=0111, oneFlag=true, ALUresult=-1, memToReg=0, writebackCLK=20, PC=16, decodeCLK=17, memWwrite=0, isImmediate=0, rdIndex=28, regDst=1, rtIndex=27, canBranch=false, branchAddress=0, opCode=1100}

ID_EX{Instruction=11001101011011110000000000000000, memRead=0, JumpAddress=11111101011011110000000000000000, 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, memWwrite=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}

```

Type to enter a caption.

```
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
```

Type to enter a caption.

```
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
```

Type to enter a caption.

```
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
```

TYPE TO ENTER A CAPTION.