**Report**

**CA Project**

Team: QeskilyahDeLaboih

Abdallah Ragab Abdelazim 43-14849 T-09

Eslam Sabry 43-15786 T-15

Khaled Elshabrawy 43-14430 T-09

Mohamed Fathy 43-17605 T-11

Ahmed Al Arousi 43-5275 T-09

**Contents**

- Microarchitecture Features

- Data-path

- Code Structure

- Output of Tests

**I-Microarchitecture Features**

### Harvard architecture

a) It has two separate memories for instructions and data.

b) It has two sets of address/data buses between CPU and memory.

c) It allows TWO simultaneous memory fetches. One fetch for the data and the other for the instructions.

**Instruction Memory and Data Memory Size**

1024 x 16-bit

**Total Number of Registers and their size**

16 General-Purpose Registers. Each is 16-bit. Program Counter has a size of 10 bits.

|  |  |  |
| --- | --- | --- |
| Register Number | name | purpose |
| 0 | $0 | hard wired zero |
| 1 | $1 | designated for lw and store |
| 2-3 | $2 , $3 | reserved for OS kernel |
| 4-8 | $4 , $5,$6 , $7 , $8 | temp registers |
| 9-11 | $9 , $10 , &11 | save registers |
| 12 | $12 | reserved for assembler |
| 13 | $13 | global pointer |
| 14 | $14 | stack pointer |
| 15 | $15 | frame pointer |

**Instruction Format**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **R-Type** |  |  |  |  |  |  |  |
| opcode | | rd | | rs | | rt | |
| 0 | 3 | 4 | 7 | 8 | 11 | 12 | 15 |

|  |
| --- |
| **I-Type** |
| opcode | | rd | | Immediate | | | |
| 0 | 3 | 4 | 7 | 8 |  |  | 15 |

|  |
| --- |
| **J-Type** |
| opcode | | Reserved | | Address | | | | |
| 0 | 3 | 4 | 5 | 6 |  |  |  | 15 |

**Instruction Set**

**a) Arithmetic Instructions:**

1. Add.
2. Add immediate.
3. Sub.
4. Multiply.

**b) Logical Instructions:**

1. And.
2. Or immediate.
3. Shift left logical.
4. Shift right logical.

**c) Data Transfer Instructions:**

1. Load word.
2. Store word.

**d) Conditional Branch Instructions:**

1. Branch on not equal.
2. Branch on greater than.

**e) Comparison Instructions:**

1. Set on less than.

**f) Unconditional Jump Instructions:**

1. Jump.

### Data Path(Cache specifications is here)

a) Instruction Memory.

b) Data Memory + Direct-mapped **Cache** (the **replacement policy** is built-in because **cache** line **replacement** is controlled by the (virtual or physical) memory address).

c) Register File.

d) Program Counter.

e) Arithmetic Logic Unit.

f) Control Unit(s).

### Pipelining Stages

a) Fetch.

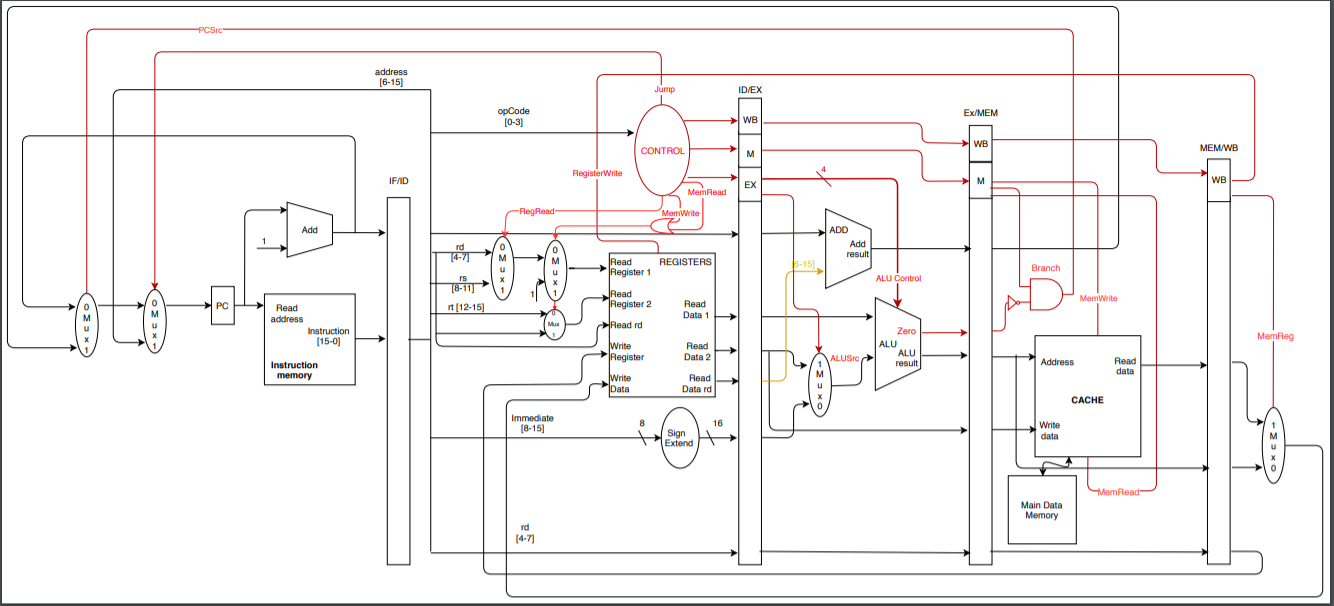
b) Decode.

c) Execute.

d) Memory.

e) Writeback.

**II-Data-path**

****

Here in the graphical representation all the 6 components needed are presented: Instruction Memory, Data Memory, Cache, Register File, Program Counter, Arithmetic Logic Unit and Control Unit.

We implemented pipelining on this microarchitecture, on each time slice (clock cycle) there are 5 simultaneous operations being processed in a different phase at most.

\*Refer to Pipeline Stages in Previous Page.

\*Pipeline stages will be explained shortly, please refer to the table at the end of the pipeline stages’ paragraph

### Pipeline Stages

### Fetch

PC module keeps track of the current instruction needed to be executed then it feeds the 10-bit address to the instruction memory which hence gets the instruction at this address, the instruction memory outputs the instruction at this address to the first pipeline registers set IF/ID and pc is added by 1 and fetched to IF/ID simultaneously.

PC takes the current address depending on 2 factors both branch and jump , they are controlled using 2 muxes both depends on 2 signals jump, branch if one of them is set , pc will be updated with the new value otherwise it will be updated with pc+1.

### Decode

Instruction is taken from IF/ID and it’s broken into parts to be checked , the least significant 4 bits resemble the op code (which is the identifier of the instruction) are fetched to the Control Unit and it accordingly generates 12 signals that control all the components.

Entries for register File depend on RegRead, MemWrite and MemRead:

R-type -> Read reg1=rs, Read reg2=rt , Read rd=rd

I-type ->Non memory: Read reg1=rd , Read reg2=rt(don’t care) , Read rd=rd

Memory(lw, sw): Read reg1:$1, Read reg2=rd, Read rd=rd

J-type->Don’t care

Also the least significant 8 bits are extended to 16 bits, and all those outputs are fetched to the ID/EX pipeline registers set (signals, Read data1, Read data2, Read data rd, rd, signEx, pc).

### Execute

The ALU takes 2 operands and a 4-bit controller taken from ID/EX(cont unit previously) and it outputs the value accordingly and a zero bit which is high if output is 0 and low otherwise.

First operand is Read data1 and the second operand is either Read data2 or signEx depending on ALUSrc.

Also pc is added with the least significant 10 bits to calculate next pc in case it’s a branch instruction. All these values are taken from ID/EX, and the outputs are fetched to EX/MEM (signals, branch address result, zero flag, alu result, rd, Read data2)

### Memory Access

The zero output is negated and anded with branch signal and used as control for the mux before the pc, in case of a branch (bne/bgt) instruction.

Remark:

* Branch not equal: ALU performs subtraction operation.
  + If the condition is true (rs!=rt), ALUResult != 0 so the zero flag is 0. We negate the zero flag (1) , perform an and operation on it with the branch signal (true) to get the PCSrc signal which is now a 1.So we update the PC to be PC+BranchAddress.
  + If the condition is false (rs == rt), ALUResult = 0 so the zero flag is 1. We negate the zero flag (0), perform an and operation on it with the branch signal (true) to get the PCSrc signal which is now a 0. So we do not update the PC.
* Branch greater than: ALU performs SetGreaterThan operation.
  + If the condition is true (rs>rt), ALUResult = 1 so the zero flag is 0. We negate the zero flag (1) , perform an and operation on it with the branch signal (true) to get the PCSrc signal which is now a 1.So we update the PC to be PC+BranchAddress.
  + If the condition is false (rs<= rt), ALUResult = 0 so the zero flag is 1. We negate the zero flag (0), perform an and operation on it with the branch signal (true) to get the PCSrc signal which is now a 0. So we do not update the PC.

Address and write data are fetched to cache and according do memwrite and memread flag it acts, if it’s a write then it’s a sw and it’s written to cache in that address, if it’s a read then it’s lw so Read data of the cache is the value of the fetched address, cache handles getting the address but if it’s a miss it accesses the main memory.

All of the inputs are fetched from the EX/MEM pipeline registers set, and the outputs are fetched to MEM/WB (signals, pc, aluresult, read data from cache and rd)

### Write Back

Register rd is fetched to Write register in register file, and either Read data from cache or alu result is written in Write data in the register file depending on MemReg flag.

If it’s a sw we take the Read data from cache otherwise we take alu result (writing in register file depends on the RegWrite flag.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Operation | opCode | Jump | Branch | RegRead | ALUSrc | MemRead | memWrite | memReg | RegWrite | ALUControl |
| Add | "0000" | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | "0000" |
| sub | "0001" | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | "0001" |
| Mult | "0010" | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | "0111" |
| AND | "0011" | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | "0010" |
| BGT | "0100" | 0 | 1 | 1 | 1 | 0 | 0 | x | 0 | "1000" |
| BNE | "0101" | 0 | 1 | 1 | 1 | 0 | 0 | x | 0 | "0001" |
| SLT | "0110" | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | "0110" |
| Addi | "0111" | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | "0000" |
| ORi | "1000" | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | "0011" |
| SLL | "1001" | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | "0100" |
| SRL | "1010" | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | "0101" |
| Lw | "1011" | 0 | 0 | x | 0 | 1 | 0 | 1 | 1 | "0000" |
| Sw | "1100" | 0 | 0 | x | 0 | 0 | 1 | x | 0 | "0000" |
| Li | "1101" | 0 | 0 | x | 0 | 0 | 0 | 0 | 1 | "1001" |
| J | "1110" | 1 | x | x | x | x | x | x | x | “xxxx” |

R-type : Add,sub,Mult,AND,BGT,BNE and SLT

I-type : Addi,ORi,SLL,SRL,lw,sw,li

J-type : jump

**III-Code Structure**

In the java project we have 4 packages we start with the package Main, classes Decoder and Encoder are just for visual presentation and class Helper is like a factory that contains the helper methods that are not of great importance.

It starts with class Main , we have instances of the 5 stages and the the 4 pipeline registers’ sets we loop n+4 times , where n is the number of instructions currently occupying the instruction memory . We apply the 5 stages simultaneously, but each takes its input from the previous pipeline registers set, as in decoding it takes its inputs from IF/ID , executing takes its inputs from ID/Ex... etc.

After the stages are done, we update the pipeline registers sets in the method updateThoseBabies(), to have the new values fetched to each of them from the previous stage to be used in the next cycle.

Then each of the 5 methods in the loop goes to its corresponding class, and all of them reside in package Stages, also pipeline registers’ sets’ classes reside in package PipelineRegisters.

In the package Modules we have the PC, RegisterFile and both MainMemory and Cache (resembling the data memory). We seek the cache first in memory access instructions if it’s a cache hit no problem if it’s a miss it goes to the main memory to fetch the data.

**IV-Output of Tests**

This is our sample :

li $1,0

li $2,5

li $3,7

addi $6,5

li $4,7

lw $7,0

li $1,4

add $8,$2,$3

ori $4,15

li $10,5

sw $2,0

nop

nop

bne $10,$10,$0

Output:

Cycle: 1

Fetch Stage:

\*Inputs:

PC= 0000000000(0)

Outputs:

Instruction: 1101000100000000

Next PC: 0000000001(1)

Decode Stage:

Execute Stage:

Memory Acess Stage:

Write Back Stage:

~~~ Cycle 1 Ended ~~~

Register File:

$0: 0000000000000000(0) $1: 0000000000000000(0) $2: 0000000000000000(0) $3: 0000000000000000(0)

$4: 0000000000000000(0) $5: 0000000000000000(0) $6: 0000000000000000(0) $7: 0000000000000000(0)

$8: 0000000000000000(0) $9: 0000000000000000(0) $10: 0000000000000000(0) $11: 0000000000000000(0)

$12: 0000000000000000(0) $13: 0000000000000000(0) $14: 0000000000000000(0) $15: 0000000000000000(0)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cycle: 2

Fetch Stage:

\*Inputs:

PC= 0000000001(1)

Outputs:

Instruction: 1101001000000101

Next PC: 0000000010(2)

Decode Stage:

Inputs:

Instruction: 1101000100000000

PC: 0000000001(1)

Outputs:

Jump Address: 0100000000(256)

Jump= 0

ID Controls: RegRead=0

EX Controls: ALU Control= 1001(9) ALUSrc= 0

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

WB Controls: MemReg= 0 RegWrite= 1

Read Data 1= 0000000000000000(0)

Read Data 2= 0000000000000000(0)

Read Branch= 0000000000(0)

Sign Extended= 0000000000000000 (0)

Write Register(rd) = 0001(1)

Execute Stage:

Memory Acess Stage:

Write Back Stage:

~~~ Cycle 2 Ended ~~~

Register File:

$0: 0000000000000000(0) $1: 0000000000000000(0) $2: 0000000000000000(0) $3: 0000000000000000(0)

$4: 0000000000000000(0) $5: 0000000000000000(0) $6: 0000000000000000(0) $7: 0000000000000000(0)

$8: 0000000000000000(0) $9: 0000000000000000(0) $10: 0000000000000000(0) $11: 0000000000000000(0)

$12: 0000000000000000(0) $13: 0000000000000000(0) $14: 0000000000000000(0) $15: 0000000000000000(0)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cycle: 3

Fetch Stage:

\*Inputs:

PC= 0000000010(2)

Outputs:

Instruction: 1101001100000111

Next PC: 0000000011(3)

Decode Stage:

Inputs:

Instruction: 1101001000000101

PC: 0000000010(2)

Outputs:

Jump Address: 1000000101(517)

Jump= 0

ID Controls: RegRead=0

EX Controls: ALU Control= 1001(9) ALUSrc= 0

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

WB Controls: MemReg= 0 RegWrite= 1

Read Data 1= 0000000000000000(0)

Read Data 2= 0000000000000000(0)

Read Branch= 0000000000(0)

Sign Extended= 0000000000000101 (5)

Write Register(rd) = 0010(2)

Execute Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

EX (This Stage) Controls: ALU Control= 1001(9) ALUSrc= 0

PC= 0000000001(1)

Read Branch = 0000000000(0)

Read Data 1 = 0000000000000000(0)

Read Data 2 = 0000000000000000(0)

Sign Extended = 0000000000000000(0)

Write Register = 00001(1)

Outputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000000001(1)

Zero Flag= 1

ALU Result= 0000000000000000(0)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00001(1)

Memory Acess Stage:

Write Back Stage:

~~~ Cycle 3 Ended ~~~

Register File:

$0: 0000000000000000(0) $1: 0000000000000000(0) $2: 0000000000000000(0) $3: 0000000000000000(0)

$4: 0000000000000000(0) $5: 0000000000000000(0) $6: 0000000000000000(0) $7: 0000000000000000(0)

$8: 0000000000000000(0) $9: 0000000000000000(0) $10: 0000000000000000(0) $11: 0000000000000000(0)

$12: 0000000000000000(0) $13: 0000000000000000(0) $14: 0000000000000000(0) $15: 0000000000000000(0)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cycle: 4

Fetch Stage:

\*Inputs:

PC= 0000000011(3)

Outputs:

Instruction: 0111011000000101

Next PC: 0000000100(4)

Decode Stage:

Inputs:

Instruction: 1101001100000111

PC: 0000000011(3)

Outputs:

Jump Address: 1100000111(775)

Jump= 0

ID Controls: RegRead=0

EX Controls: ALU Control= 1001(9) ALUSrc= 0

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

WB Controls: MemReg= 0 RegWrite= 1

Read Data 1= 0000000000000000(0)

Read Data 2= 0000000000000000(0)

Read Branch= 0000000000(0)

Sign Extended= 0000000000000111 (7)

Write Register(rd) = 0011(3)

Execute Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

EX (This Stage) Controls: ALU Control= 1001(9) ALUSrc= 0

PC= 0000000010(2)

Read Branch = 0000000000(0)

Read Data 1 = 0000000000000000(0)

Read Data 2 = 0000000000000000(0)

Sign Extended = 0000000000000101(5)

Write Register = 00010(2)

Outputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000000010(2)

Zero Flag= 0

ALU Result= 0000000000000101(5)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00010(2)

Memory Acess Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000000000000000(0)

Zero Flag= 1

ALU Result= 0000000000000000(0)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00001(1)

Outputs:

Branch Address Result= 0000000001(1)

PCSrc:0

WB Controls: MemReg= 0 RegWrite= 1

Read Data (from memory): 0000000000000000(0)

ALU Result= 0000000000000000(0)

Write Register = 00001(1)

Write Back Stage:

~~~ Cycle 4 Ended ~~~

Register File:

$0: 0000000000000000(0) $1: 0000000000000000(0) $2: 0000000000000000(0) $3: 0000000000000000(0)

$4: 0000000000000000(0) $5: 0000000000000000(0) $6: 0000000000000000(0) $7: 0000000000000000(0)

$8: 0000000000000000(0) $9: 0000000000000000(0) $10: 0000000000000000(0) $11: 0000000000000000(0)

$12: 0000000000000000(0) $13: 0000000000000000(0) $14: 0000000000000000(0) $15: 0000000000000000(0)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cycle: 5

Fetch Stage:

\*Inputs:

PC= 0000000100(4)

Outputs:

Instruction: 1101010000000111

Next PC: 0000000101(5)

Decode Stage:

Inputs:

Instruction: 0111011000000101

PC: 0000000100(4)

Outputs:

Jump Address: 1000000101(517)

Jump= 0

ID Controls: RegRead=0

EX Controls: ALU Control= 0000(0) ALUSrc= 0

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

WB Controls: MemReg= 0 RegWrite= 1

Read Data 1= 0000000000000000(0)

Read Data 2= 0000000000000000(0)

Read Branch= 0000000000(0)

Sign Extended= 0000000000000101 (5)

Write Register(rd) = 0110(6)

Execute Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

EX (This Stage) Controls: ALU Control= 1001(9) ALUSrc= 0

PC= 0000000011(3)

Read Branch = 0000000000(0)

Read Data 1 = 0000000000000000(0)

Read Data 2 = 0000000000000000(0)

Sign Extended = 0000000000000111(7)

Write Register = 00011(3)

Outputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000000011(3)

Zero Flag= 0

ALU Result= 0000000000000111(7)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00011(3)

Memory Acess Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000000001(1)

Zero Flag= 0

ALU Result= 0000000000000101(5)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00010(2)

Outputs:

Branch Address Result= 0000000010(2)

PCSrc:0

WB Controls: MemReg= 0 RegWrite= 1

Read Data (from memory): 0000000000000000(0)

ALU Result= 0000000000000101(5)

Write Register = 00010(2)

Write Back Stage:

Inputs:

WB Controls: RegWrite= 1

MemReg= 0 ALU Result= 0000000000000000(0)

Read Data (from memory): 0000000000000000(0)

Write Register = 00001(1)

Outputs:

RegFile Before Writing: [0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

RegFile After Writing: [0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

~~~ Cycle 5 Ended ~~~

Register File:

$0: 0000000000000000(0) $1: 0000000000000000(0) $2: 0000000000000000(0) $3: 0000000000000000(0)

$4: 0000000000000000(0) $5: 0000000000000000(0) $6: 0000000000000000(0) $7: 0000000000000000(0)

$8: 0000000000000000(0) $9: 0000000000000000(0) $10: 0000000000000000(0) $11: 0000000000000000(0)

$12: 0000000000000000(0) $13: 0000000000000000(0) $14: 0000000000000000(0) $15: 0000000000000000(0)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cycle: 6

Fetch Stage:

\*Inputs:

PC= 0000000101(5)

Outputs:

Instruction: 1011011100000000

Next PC: 0000000110(6)

Decode Stage:

Inputs:

Instruction: 1101010000000111

PC: 0000000101(5)

Outputs:

Jump Address: 0000000111(7)

Jump= 0

ID Controls: RegRead=0

EX Controls: ALU Control= 1001(9) ALUSrc= 0

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

WB Controls: MemReg= 0 RegWrite= 1

Read Data 1= 0000000000000000(0)

Read Data 2= 0000000000000000(0)

Read Branch= 0000000000(0)

Sign Extended= 0000000000000111 (7)

Write Register(rd) = 0100(4)

Execute Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

EX (This Stage) Controls: ALU Control= 0000(0) ALUSrc= 0

PC= 0000000100(4)

Read Branch = 0000000000(0)

Read Data 1 = 0000000000000000(0)

Read Data 2 = 0000000000000000(0)

Sign Extended = 0000000000000101(5)

Write Register = 00110(6)

Outputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000000100(4)

Zero Flag= 0

ALU Result= 0000000000000101(5)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00110(6)

Memory Acess Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000000010(2)

Zero Flag= 0

ALU Result= 0000000000000111(7)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00011(3)

Outputs:

Branch Address Result= 0000000011(3)

PCSrc:0

WB Controls: MemReg= 0 RegWrite= 1

Read Data (from memory): 0000000000000000(0)

ALU Result= 0000000000000111(7)

Write Register = 00011(3)

Write Back Stage:

Inputs:

WB Controls: RegWrite= 1

MemReg= 0 ALU Result= 0000000000000101(5)

Read Data (from memory): 0000000000000000(0)

Write Register = 00010(2)

Outputs:

RegFile Before Writing: [0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

RegFile After Writing: [0000000000000000, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

~~~ Cycle 6 Ended ~~~

Register File:

$0: 0000000000000000(0) $1: 0000000000000000(0) $2: 0000000000000101(5) $3: 0000000000000000(0)

$4: 0000000000000000(0) $5: 0000000000000000(0) $6: 0000000000000000(0) $7: 0000000000000000(0)

$8: 0000000000000000(0) $9: 0000000000000000(0) $10: 0000000000000000(0) $11: 0000000000000000(0)

$12: 0000000000000000(0) $13: 0000000000000000(0) $14: 0000000000000000(0) $15: 0000000000000000(0)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cycle: 7

Fetch Stage:

\*Inputs:

PC= 0000000110(6)

Outputs:

Instruction: 1101000100000100

Next PC: 0000000111(7)

Decode Stage:

Inputs:

Instruction: 1011011100000000

PC: 0000000110(6)

Outputs:

Jump Address: 1100000000(768)

Jump= 0

ID Controls: RegRead=0

EX Controls: ALU Control= 0000(0) ALUSrc= 0

MEM Controls: MemRead= 1 MemWrite= 0 Branch= 0

WB Controls: MemReg= 1 RegWrite= 1

Read Data 1= 0000000000000000(0)

Read Data 2= 0000000000000000(0)

Read Branch= 0000000000(0)

Sign Extended= 0000000000000000 (0)

Write Register(rd) = 0111(7)

Execute Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

EX (This Stage) Controls: ALU Control= 1001(9) ALUSrc= 0

PC= 0000000101(5)

Read Branch = 0000000000(0)

Read Data 1 = 0000000000000000(0)

Read Data 2 = 0000000000000000(0)

Sign Extended = 0000000000000111(7)

Write Register = 00100(4)

Outputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000000101(5)

Zero Flag= 0

ALU Result= 0000000000000111(7)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00100(4)

Memory Acess Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000000011(3)

Zero Flag= 0

ALU Result= 0000000000000101(5)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00110(6)

Outputs:

Branch Address Result= 0000000100(4)

PCSrc:0

WB Controls: MemReg= 0 RegWrite= 1

Read Data (from memory): 0000000000000000(0)

ALU Result= 0000000000000101(5)

Write Register = 00110(6)

Write Back Stage:

Inputs:

WB Controls: RegWrite= 1

MemReg= 0 ALU Result= 0000000000000111(7)

Read Data (from memory): 0000000000000000(0)

Write Register = 00011(3)

Outputs:

RegFile Before Writing: [0000000000000000, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

RegFile After Writing: [0000000000000000, 0000000000000000, 0000000000000101, 0000000000000111, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

~~~ Cycle 7 Ended ~~~

Register File:

$0: 0000000000000000(0) $1: 0000000000000000(0) $2: 0000000000000101(5) $3: 0000000000000111(7)

$4: 0000000000000000(0) $5: 0000000000000000(0) $6: 0000000000000000(0) $7: 0000000000000000(0)

$8: 0000000000000000(0) $9: 0000000000000000(0) $10: 0000000000000000(0) $11: 0000000000000000(0)

$12: 0000000000000000(0) $13: 0000000000000000(0) $14: 0000000000000000(0) $15: 0000000000000000(0)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cycle: 8

Fetch Stage:

\*Inputs:

PC= 0000000111(7)

Outputs:

Instruction: 0000100000100011

Next PC: 0000001000(8)

Decode Stage:

Inputs:

Instruction: 1101000100000100

PC: 0000000111(7)

Outputs:

Jump Address: 0100000100(260)

Jump= 0

ID Controls: RegRead=0

EX Controls: ALU Control= 1001(9) ALUSrc= 0

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

WB Controls: MemReg= 0 RegWrite= 1

Read Data 1= 0000000000000000(0)

Read Data 2= 0000000000000000(0)

Read Branch= 0000000000(0)

Sign Extended= 0000000000000100 (4)

Write Register(rd) = 0001(1)

Execute Stage:

Inputs:

WB Controls: MemReg= 1 RegWrite= 1

MEM Controls: MemRead= 1 MemWrite= 0 Branch= 0

EX (This Stage) Controls: ALU Control= 0000(0) ALUSrc= 0

PC= 0000000110(6)

Read Branch = 0000000000(0)

Read Data 1 = 0000000000000000(0)

Read Data 2 = 0000000000000000(0)

Sign Extended = 0000000000000000(0)

Write Register = 00111(7)

Outputs:

WB Controls: MemReg= 1 RegWrite= 1

MEM Controls: MemRead= 1 MemWrite= 0 Branch= 0

Branch Address Result= 0000000110(6)

Zero Flag= 1

ALU Result= 0000000000000000(0)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00111(7)

Memory Acess Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000000100(4)

Zero Flag= 0

ALU Result= 0000000000000111(7)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00100(4)

Outputs:

Branch Address Result= 0000000101(5)

PCSrc:0

WB Controls: MemReg= 0 RegWrite= 1

Read Data (from memory): 0000000000000000(0)

ALU Result= 0000000000000111(7)

Write Register = 00100(4)

Write Back Stage:

Inputs:

WB Controls: RegWrite= 1

MemReg= 0 ALU Result= 0000000000000101(5)

Read Data (from memory): 0000000000000000(0)

Write Register = 00110(6)

Outputs:

RegFile Before Writing: [0000000000000000, 0000000000000000, 0000000000000101, 0000000000000111, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

RegFile After Writing: [0000000000000000, 0000000000000000, 0000000000000101, 0000000000000111, 0000000000000000, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

~~~ Cycle 8 Ended ~~~

Register File:

$0: 0000000000000000(0) $1: 0000000000000000(0) $2: 0000000000000101(5) $3: 0000000000000111(7)

$4: 0000000000000000(0) $5: 0000000000000000(0) $6: 0000000000000101(5) $7: 0000000000000000(0)

$8: 0000000000000000(0) $9: 0000000000000000(0) $10: 0000000000000000(0) $11: 0000000000000000(0)

$12: 0000000000000000(0) $13: 0000000000000000(0) $14: 0000000000000000(0) $15: 0000000000000000(0)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cycle: 9

Fetch Stage:

\*Inputs:

PC= 0000001000(8)

Outputs:

Instruction: 1000010000001111

Next PC: 0000001001(9)

Decode Stage:

Inputs:

Instruction: 0000100000100011

PC: 0000001000(8)

Outputs:

Jump Address: 0000100011(35)

Jump= 0

ID Controls: RegRead=1

EX Controls: ALU Control= 0000(0) ALUSrc= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

WB Controls: MemReg= 0 RegWrite= 1

Read Data 1= 0000000000000101(5)

Read Data 2= 0000000000000111(7)

Read Branch= 0000000000(0)

Sign Extended= 0000000000100011 (35)

Write Register(rd) = 1000(8)

Execute Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

EX (This Stage) Controls: ALU Control= 1001(9) ALUSrc= 0

PC= 0000000111(7)

Read Branch = 0000000000(0)

Read Data 1 = 0000000000000000(0)

Read Data 2 = 0000000000000000(0)

Sign Extended = 0000000000000100(4)

Write Register = 00001(1)

Outputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000000111(7)

Zero Flag= 0

ALU Result= 0000000000000100(4)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00001(1)

Memory Acess Stage:

Inputs:

WB Controls: MemReg= 1 RegWrite= 1

MEM Controls: MemRead= 1 MemWrite= 0 Branch= 0

Branch Address Result= 0000000101(5)

Zero Flag= 1

ALU Result= 0000000000000000(0)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00111(7)

Outputs:

Branch Address Result= 0000000110(6)

PCSrc:0

I am the cache | READ OPERATION

Miss

WB Controls: MemReg= 1 RegWrite= 1

Read Data (from memory): 0000000000000000(0)

ALU Result= 0000000000000000(0)

Write Register = 00111(7)

Write Back Stage:

Inputs:

WB Controls: RegWrite= 1

MemReg= 0 ALU Result= 0000000000000111(7)

Read Data (from memory): 0000000000000000(0)

Write Register = 00100(4)

Outputs:

RegFile Before Writing: [0000000000000000, 0000000000000000, 0000000000000101, 0000000000000111, 0000000000000000, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

RegFile After Writing: [0000000000000000, 0000000000000000, 0000000000000101, 0000000000000111, 0000000000000111, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

~~~ Cycle 9 Ended ~~~

Register File:

$0: 0000000000000000(0) $1: 0000000000000000(0) $2: 0000000000000101(5) $3: 0000000000000111(7)

$4: 0000000000000111(7) $5: 0000000000000000(0) $6: 0000000000000101(5) $7: 0000000000000000(0)

$8: 0000000000000000(0) $9: 0000000000000000(0) $10: 0000000000000000(0) $11: 0000000000000000(0)

$12: 0000000000000000(0) $13: 0000000000000000(0) $14: 0000000000000000(0) $15: 0000000000000000(0)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cycle: 10

Fetch Stage:

\*Inputs:

PC= 0000001001(9)

Outputs:

Instruction: 1101101000000101

Next PC: 0000001010(10)

Decode Stage:

Inputs:

Instruction: 1000010000001111

PC: 0000001001(9)

Outputs:

Jump Address: 0000001111(15)

Jump= 0

ID Controls: RegRead=0

EX Controls: ALU Control= 0011(3) ALUSrc= 0

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

WB Controls: MemReg= 0 RegWrite= 1

Read Data 1= 0000000000000111(7)

Read Data 2= 0000000000000000(0)

Read Branch= 0000000111(7)

Sign Extended= 0000000000001111 (15)

Write Register(rd) = 0100(4)

Execute Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

EX (This Stage) Controls: ALU Control= 0000(0) ALUSrc= 1

PC= 0000001000(8)

Read Branch = 0000000000(0)

Read Data 1 = 0000000000000101(5)

Read Data 2 = 0000000000000111(7)

Sign Extended = 0000000000100011(35)

Write Register = 01000(8)

Outputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000001000(8)

Zero Flag= 0

ALU Result= 0000000000001100(12)

Read Data 2 (Value to write in memory if SW) = 0000000000000111(7)

Write Register = 01000(8)

Memory Acess Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000000110(6)

Zero Flag= 0

ALU Result= 0000000000000100(4)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00001(1)

Outputs:

Branch Address Result= 0000000111(7)

PCSrc:0

WB Controls: MemReg= 0 RegWrite= 1

Read Data (from memory): 0000000000000000(0)

ALU Result= 0000000000000100(4)

Write Register = 00001(1)

Write Back Stage:

Inputs:

WB Controls: RegWrite= 1

MemReg= 1 ALU Result= 0000000000000000(0)

Read Data (from memory): 0000000000000000(0)

Write Register = 00111(7)

Outputs:

RegFile Before Writing: [0000000000000000, 0000000000000000, 0000000000000101, 0000000000000111, 0000000000000111, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

RegFile After Writing: [0000000000000000, 0000000000000000, 0000000000000101, 0000000000000111, 0000000000000111, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

~~~ Cycle 10 Ended ~~~

Register File:

$0: 0000000000000000(0) $1: 0000000000000000(0) $2: 0000000000000101(5) $3: 0000000000000111(7)

$4: 0000000000000111(7) $5: 0000000000000000(0) $6: 0000000000000101(5) $7: 0000000000000000(0)

$8: 0000000000000000(0) $9: 0000000000000000(0) $10: 0000000000000000(0) $11: 0000000000000000(0)

$12: 0000000000000000(0) $13: 0000000000000000(0) $14: 0000000000000000(0) $15: 0000000000000000(0)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cycle: 11

Fetch Stage:

\*Inputs:

PC= 0000001010(10)

Outputs:

Instruction: 1100001000000000

Next PC: 0000001011(11)

Decode Stage:

Inputs:

Instruction: 1101101000000101

PC: 0000001010(10)

Outputs:

Jump Address: 1000000101(517)

Jump= 0

ID Controls: RegRead=0

EX Controls: ALU Control= 1001(9) ALUSrc= 0

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

WB Controls: MemReg= 0 RegWrite= 1

Read Data 1= 0000000000000000(0)

Read Data 2= 0000000000000000(0)

Read Branch= 0000000000(0)

Sign Extended= 0000000000000101 (5)

Write Register(rd) = 1010(10)

Execute Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

EX (This Stage) Controls: ALU Control= 0011(3) ALUSrc= 0

PC= 0000001001(9)

Read Branch = 0000000111(7)

Read Data 1 = 0000000000000111(7)

Read Data 2 = 0000000000000000(0)

Sign Extended = 0000000000001111(15)

Write Register = 00100(4)

Outputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000010000(16)

Zero Flag= 0

ALU Result= 0000000000001111(15)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00100(4)

Memory Acess Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000000111(7)

Zero Flag= 0

ALU Result= 0000000000001100(12)

Read Data 2 (Value to write in memory if SW) = 0000000000000111(7)

Write Register = 01000(8)

Outputs:

Branch Address Result= 0000001000(8)

PCSrc:0

WB Controls: MemReg= 0 RegWrite= 1

Read Data (from memory): 0000000000000000(0)

ALU Result= 0000000000001100(12)

Write Register = 01000(8)

Write Back Stage:

Inputs:

WB Controls: RegWrite= 1

MemReg= 0 ALU Result= 0000000000000100(4)

Read Data (from memory): 0000000000000000(0)

Write Register = 00001(1)

Outputs:

RegFile Before Writing: [0000000000000000, 0000000000000000, 0000000000000101, 0000000000000111, 0000000000000111, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

RegFile After Writing: [0000000000000000, 0000000000000100, 0000000000000101, 0000000000000111, 0000000000000111, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

~~~ Cycle 11 Ended ~~~

Register File:

$0: 0000000000000000(0) $1: 0000000000000100(4) $2: 0000000000000101(5) $3: 0000000000000111(7)

$4: 0000000000000111(7) $5: 0000000000000000(0) $6: 0000000000000101(5) $7: 0000000000000000(0)

$8: 0000000000000000(0) $9: 0000000000000000(0) $10: 0000000000000000(0) $11: 0000000000000000(0)

$12: 0000000000000000(0) $13: 0000000000000000(0) $14: 0000000000000000(0) $15: 0000000000000000(0)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cycle: 12

Fetch Stage:

\*Inputs:

PC= 0000001011(11)

Outputs:

Instruction: 0000000000000000

Next PC: 0000001100(12)

Decode Stage:

Inputs:

Instruction: 1100001000000000

PC: 0000001011(11)

Outputs:

Jump Address: 1000000000(512)

Jump= 0

ID Controls: RegRead=0

EX Controls: ALU Control= 0000(0) ALUSrc= 0

MEM Controls: MemRead= 0 MemWrite= 1 Branch= 0

WB Controls: MemReg= 0 RegWrite= 0

Read Data 1= 0000000000000100(4)

Read Data 2= 0000000000000101(5)

Read Branch= 0000000101(5)

Sign Extended= 0000000000000000 (0)

Write Register(rd) = 0010(2)

Execute Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

EX (This Stage) Controls: ALU Control= 1001(9) ALUSrc= 0

PC= 0000001010(10)

Read Branch = 0000000000(0)

Read Data 1 = 0000000000000000(0)

Read Data 2 = 0000000000000000(0)

Sign Extended = 0000000000000101(5)

Write Register = 01010(10)

Outputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000001010(10)

Zero Flag= 0

ALU Result= 0000000000000101(5)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 01010(10)

Memory Acess Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000001000(8)

Zero Flag= 0

ALU Result= 0000000000001111(15)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00100(4)

Outputs:

Branch Address Result= 0000010000(16)

PCSrc:0

WB Controls: MemReg= 0 RegWrite= 1

Read Data (from memory): 0000000000000000(0)

ALU Result= 0000000000001111(15)

Write Register = 00100(4)

Write Back Stage:

Inputs:

WB Controls: RegWrite= 1

MemReg= 0 ALU Result= 0000000000001100(12)

Read Data (from memory): 0000000000000000(0)

Write Register = 01000(8)

Outputs:

RegFile Before Writing: [0000000000000000, 0000000000000100, 0000000000000101, 0000000000000111, 0000000000000111, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

RegFile After Writing: [0000000000000000, 0000000000000100, 0000000000000101, 0000000000000111, 0000000000000111, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000001100, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

~~~ Cycle 12 Ended ~~~

Register File:

$0: 0000000000000000(0) $1: 0000000000000100(4) $2: 0000000000000101(5) $3: 0000000000000111(7)

$4: 0000000000000111(7) $5: 0000000000000000(0) $6: 0000000000000101(5) $7: 0000000000000000(0)

$8: 0000000000001100(12) $9: 0000000000000000(0) $10: 0000000000000000(0) $11: 0000000000000000(0)

$12: 0000000000000000(0) $13: 0000000000000000(0) $14: 0000000000000000(0) $15: 0000000000000000(0)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cycle: 13

Fetch Stage:

\*Inputs:

PC= 0000001100(12)

Outputs:

Instruction: 0000000000000000

Next PC: 0000001101(13)

Decode Stage:

Inputs:

Instruction: 0000000000000000

PC: 0000001100(12)

Outputs:

Jump Address: 0000000000(0)

Jump= 0

ID Controls: RegRead=1

EX Controls: ALU Control= 0000(0) ALUSrc= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

WB Controls: MemReg= 0 RegWrite= 1

Read Data 1= 0000000000000000(0)

Read Data 2= 0000000000000000(0)

Read Branch= 0000000000(0)

Sign Extended= 0000000000000000 (0)

Write Register(rd) = 0000(0)

Execute Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 0

MEM Controls: MemRead= 0 MemWrite= 1 Branch= 0

EX (This Stage) Controls: ALU Control= 0000(0) ALUSrc= 0

PC= 0000001011(11)

Read Branch = 0000000101(5)

Read Data 1 = 0000000000000100(4)

Read Data 2 = 0000000000000101(5)

Sign Extended = 0000000000000000(0)

Write Register = 00010(2)

Outputs:

WB Controls: MemReg= 0 RegWrite= 0

MEM Controls: MemRead= 0 MemWrite= 1 Branch= 0

Branch Address Result= 0000010000(16)

Zero Flag= 0

ALU Result= 0000000000000100(4)

Read Data 2 (Value to write in memory if SW) = 0000000000000101(5)

Write Register = 00010(2)

Memory Acess Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000010000(16)

Zero Flag= 0

ALU Result= 0000000000000101(5)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 01010(10)

Outputs:

Branch Address Result= 0000001010(10)

PCSrc:0

WB Controls: MemReg= 0 RegWrite= 1

Read Data (from memory): 0000000000000000(0)

ALU Result= 0000000000000101(5)

Write Register = 01010(10)

Write Back Stage:

Inputs:

WB Controls: RegWrite= 1

MemReg= 0 ALU Result= 0000000000001111(15)

Read Data (from memory): 0000000000000000(0)

Write Register = 00100(4)

Outputs:

RegFile Before Writing: [0000000000000000, 0000000000000100, 0000000000000101, 0000000000000111, 0000000000000111, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000001100, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

RegFile After Writing: [0000000000000000, 0000000000000100, 0000000000000101, 0000000000000111, 0000000000001111, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000001100, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

~~~ Cycle 13 Ended ~~~

Register File:

$0: 0000000000000000(0) $1: 0000000000000100(4) $2: 0000000000000101(5) $3: 0000000000000111(7)

$4: 0000000000001111(15) $5: 0000000000000000(0) $6: 0000000000000101(5) $7: 0000000000000000(0)

$8: 0000000000001100(12) $9: 0000000000000000(0) $10: 0000000000000000(0) $11: 0000000000000000(0)

$12: 0000000000000000(0) $13: 0000000000000000(0) $14: 0000000000000000(0) $15: 0000000000000000(0)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cycle: 14

Fetch Stage:

\*Inputs:

PC= 0000001101(13)

Outputs:

Instruction: 0101101010100000

Next PC: 0000001110(14)

Decode Stage:

Inputs:

Instruction: 0000000000000000

PC: 0000001101(13)

Outputs:

Jump Address: 0000000000(0)

Jump= 0

ID Controls: RegRead=1

EX Controls: ALU Control= 0000(0) ALUSrc= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

WB Controls: MemReg= 0 RegWrite= 1

Read Data 1= 0000000000000000(0)

Read Data 2= 0000000000000000(0)

Read Branch= 0000000000(0)

Sign Extended= 0000000000000000 (0)

Write Register(rd) = 0000(0)

Execute Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

EX (This Stage) Controls: ALU Control= 0000(0) ALUSrc= 1

PC= 0000001100(12)

Read Branch = 0000000000(0)

Read Data 1 = 0000000000000000(0)

Read Data 2 = 0000000000000000(0)

Sign Extended = 0000000000000000(0)

Write Register = 00000(0)

Outputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000001100(12)

Zero Flag= 1

ALU Result= 0000000000000000(0)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00000(0)

Memory Acess Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 0

MEM Controls: MemRead= 0 MemWrite= 1 Branch= 0

Branch Address Result= 0000001010(10)

Zero Flag= 0

ALU Result= 0000000000000100(4)

Read Data 2 (Value to write in memory if SW) = 0000000000000101(5)

Write Register = 00010(2)

Outputs:

Branch Address Result= 0000010000(16)

PCSrc:0

I am the cache; Write Operation |

offset=4, tag=0, data=0000000000000101

I am the Data Memory | Write Operation

dataMemory[4]=0000000000000101

WB Controls: MemReg= 0 RegWrite= 0

Read Data (from memory): 0000000000000000(0)

ALU Result= 0000000000000100(4)

Write Register = 00010(2)

Write Back Stage:

Inputs:

WB Controls: RegWrite= 1

MemReg= 0 ALU Result= 0000000000000101(5)

Read Data (from memory): 0000000000000000(0)

Write Register = 01010(10)

Outputs:

RegFile Before Writing: [0000000000000000, 0000000000000100, 0000000000000101, 0000000000000111, 0000000000001111, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000001100, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

RegFile After Writing: [0000000000000000, 0000000000000100, 0000000000000101, 0000000000000111, 0000000000001111, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000001100, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

~~~ Cycle 14 Ended ~~~

Register File:

$0: 0000000000000000(0) $1: 0000000000000100(4) $2: 0000000000000101(5) $3: 0000000000000111(7)

$4: 0000000000001111(15) $5: 0000000000000000(0) $6: 0000000000000101(5) $7: 0000000000000000(0)

$8: 0000000000001100(12) $9: 0000000000000000(0) $10: 0000000000000101(5) $11: 0000000000000000(0)

$12: 0000000000000000(0) $13: 0000000000000000(0) $14: 0000000000000000(0) $15: 0000000000000000(0)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cycle: 15

Fetch Stage:

\*Inputs:

PC= 0000001110(14)

Outputs:

Instruction: 0000000000000000

Next PC: 0000001111(15)

Decode Stage:

Inputs:

Instruction: 0101101010100000

PC: 0000001110(14)

Outputs:

Jump Address: 1010100000(672)

Jump= 0

ID Controls: RegRead=1

EX Controls: ALU Control= 0001(1) ALUSrc= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 1

WB Controls: MemReg= 0 RegWrite= 0

Read Data 1= 0000000000000101(5)

Read Data 2= 0000000000000000(0)

Read Branch= 0000000101(5)

Sign Extended= 1111111110100000 (-96)

Write Register(rd) = 1010(10)

Execute Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

EX (This Stage) Controls: ALU Control= 0000(0) ALUSrc= 1

PC= 0000001101(13)

Read Branch = 0000000000(0)

Read Data 1 = 0000000000000000(0)

Read Data 2 = 0000000000000000(0)

Sign Extended = 0000000000000000(0)

Write Register = 00000(0)

Outputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000001101(13)

Zero Flag= 1

ALU Result= 0000000000000000(0)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00000(0)

Memory Acess Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000010000(16)

Zero Flag= 1

ALU Result= 0000000000000000(0)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00000(0)

Outputs:

Branch Address Result= 0000001100(12)

PCSrc:0

WB Controls: MemReg= 0 RegWrite= 1

Read Data (from memory): 0000000000000000(0)

ALU Result= 0000000000000000(0)

Write Register = 00000(0)

Write Back Stage:

Inputs:

WB Controls: RegWrite= 0

MemReg= 0 ALU Result= 0000000000000100(4)

Read Data (from memory): 0000000000000000(0)

Write Register = 00010(2)

Outputs:

RegFile Before Writing: [0000000000000000, 0000000000000100, 0000000000000101, 0000000000000111, 0000000000001111, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000001100, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

RegFile After Writing: [0000000000000000, 0000000000000100, 0000000000000101, 0000000000000111, 0000000000001111, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000001100, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

~~~ Cycle 15 Ended ~~~

Register File:

$0: 0000000000000000(0) $1: 0000000000000100(4) $2: 0000000000000101(5) $3: 0000000000000111(7)

$4: 0000000000001111(15) $5: 0000000000000000(0) $6: 0000000000000101(5) $7: 0000000000000000(0)

$8: 0000000000001100(12) $9: 0000000000000000(0) $10: 0000000000000101(5) $11: 0000000000000000(0)

$12: 0000000000000000(0) $13: 0000000000000000(0) $14: 0000000000000000(0) $15: 0000000000000000(0)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cycle: 16

Fetch Stage:

\*Inputs:

PC= 0000001111(15)

Outputs:

Instruction: 0000000000000000

Next PC: 0000010000(16)

Decode Stage:

Inputs:

Instruction: 0000000000000000

PC: 0000001111(15)

Outputs:

Jump Address: 0000000000(0)

Jump= 0

ID Controls: RegRead=1

EX Controls: ALU Control= 0000(0) ALUSrc= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

WB Controls: MemReg= 0 RegWrite= 1

Read Data 1= 0000000000000000(0)

Read Data 2= 0000000000000000(0)

Read Branch= 0000000000(0)

Sign Extended= 0000000000000000 (0)

Write Register(rd) = 0000(0)

Execute Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 0

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 1

EX (This Stage) Controls: ALU Control= 0001(1) ALUSrc= 1

PC= 0000001110(14)

Read Branch = 0000000101(5)

Read Data 1 = 0000000000000101(5)

Read Data 2 = 0000000000000000(0)

Sign Extended = 1111111110100000(-96)

Write Register = 01010(10)

Outputs:

WB Controls: MemReg= 0 RegWrite= 0

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 1

Branch Address Result= 0000010011(19)

Zero Flag= 0

ALU Result= 0000000000000101(5)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 01010(10)

Memory Acess Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000001100(12)

Zero Flag= 1

ALU Result= 0000000000000000(0)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00000(0)

Outputs:

Branch Address Result= 0000001101(13)

PCSrc:0

WB Controls: MemReg= 0 RegWrite= 1

Read Data (from memory): 0000000000000000(0)

ALU Result= 0000000000000000(0)

Write Register = 00000(0)

Write Back Stage:

Inputs:

WB Controls: RegWrite= 1

MemReg= 0 ALU Result= 0000000000000000(0)

Read Data (from memory): 0000000000000000(0)

Write Register = 00000(0)

Outputs:

RegFile Before Writing: [0000000000000000, 0000000000000100, 0000000000000101, 0000000000000111, 0000000000001111, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000001100, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

~~~ Cycle 16 Ended ~~~

Register File:

$0: 0000000000000000(0) $1: 0000000000000100(4) $2: 0000000000000101(5) $3: 0000000000000111(7)

$4: 0000000000001111(15) $5: 0000000000000000(0) $6: 0000000000000101(5) $7: 0000000000000000(0)

$8: 0000000000001100(12) $9: 0000000000000000(0) $10: 0000000000000101(5) $11: 0000000000000000(0)

$12: 0000000000000000(0) $13: 0000000000000000(0) $14: 0000000000000000(0) $15: 0000000000000000(0)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cycle: 17

Fetch Stage:

\*Inputs:

PC= 0000010000(16)

Outputs:

Instruction: 0000000000000000

Next PC: 0000010001(17)

Decode Stage:

Inputs:

Instruction: 0000000000000000

PC: 0000010000(16)

Outputs:

Jump Address: 0000000000(0)

Jump= 0

ID Controls: RegRead=1

EX Controls: ALU Control= 0000(0) ALUSrc= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

WB Controls: MemReg= 0 RegWrite= 1

Read Data 1= 0000000000000000(0)

Read Data 2= 0000000000000000(0)

Read Branch= 0000000000(0)

Sign Extended= 0000000000000000 (0)

Write Register(rd) = 0000(0)

Execute Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

EX (This Stage) Controls: ALU Control= 0000(0) ALUSrc= 1

PC= 0000001111(15)

Read Branch = 0000000000(0)

Read Data 1 = 0000000000000000(0)

Read Data 2 = 0000000000000000(0)

Sign Extended = 0000000000000000(0)

Write Register = 00000(0)

Outputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000001111(15)

Zero Flag= 1

ALU Result= 0000000000000000(0)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00000(0)

Memory Acess Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 0

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 1

Branch Address Result= 0000001101(13)

Zero Flag= 0

ALU Result= 0000000000000101(5)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 01010(10)

Outputs:

Branch Address Result= 0000010011(19)

PCSrc:1

WB Controls: MemReg= 0 RegWrite= 0

Read Data (from memory): 0000000000000000(0)

ALU Result= 0000000000000101(5)

Write Register = 01010(10)

Write Back Stage:

Inputs:

WB Controls: RegWrite= 1

MemReg= 0 ALU Result= 0000000000000000(0)

Read Data (from memory): 0000000000000000(0)

Write Register = 00000(0)

Outputs:

RegFile Before Writing: [0000000000000000, 0000000000000100, 0000000000000101, 0000000000000111, 0000000000001111, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000001100, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

~~~ Cycle 17 Ended ~~~

Register File:

$0: 0000000000000000(0) $1: 0000000000000100(4) $2: 0000000000000101(5) $3: 0000000000000111(7)

$4: 0000000000001111(15) $5: 0000000000000000(0) $6: 0000000000000101(5) $7: 0000000000000000(0)

$8: 0000000000001100(12) $9: 0000000000000000(0) $10: 0000000000000101(5) $11: 0000000000000000(0)

$12: 0000000000000000(0) $13: 0000000000000000(0) $14: 0000000000000000(0) $15: 0000000000000000(0)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cycle: 18

Fetch Stage:

\*Inputs:

PC= 0000010011(19)

Outputs:

Instruction: 0000000000000000

Next PC: 0000010100(20)

Decode Stage:

Inputs:

Instruction: 0000000000000000

PC: 0000010001(17)

Outputs:

Jump Address: 0000000000(0)

Jump= 0

ID Controls: RegRead=1

EX Controls: ALU Control= 0000(0) ALUSrc= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

WB Controls: MemReg= 0 RegWrite= 1

Read Data 1= 0000000000000000(0)

Read Data 2= 0000000000000000(0)

Read Branch= 0000000000(0)

Sign Extended= 0000000000000000 (0)

Write Register(rd) = 0000(0)

Execute Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

EX (This Stage) Controls: ALU Control= 0000(0) ALUSrc= 1

PC= 0000010000(16)

Read Branch = 0000000000(0)

Read Data 1 = 0000000000000000(0)

Read Data 2 = 0000000000000000(0)

Sign Extended = 0000000000000000(0)

Write Register = 00000(0)

Outputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000010000(16)

Zero Flag= 1

ALU Result= 0000000000000000(0)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00000(0)

Memory Acess Stage:

Inputs:

WB Controls: MemReg= 0 RegWrite= 1

MEM Controls: MemRead= 0 MemWrite= 0 Branch= 0

Branch Address Result= 0000010011(19)

Zero Flag= 1

ALU Result= 0000000000000000(0)

Read Data 2 (Value to write in memory if SW) = 0000000000000000(0)

Write Register = 00000(0)

Outputs:

Branch Address Result= 0000001111(15)

PCSrc:0

WB Controls: MemReg= 0 RegWrite= 1

Read Data (from memory): 0000000000000000(0)

ALU Result= 0000000000000000(0)

Write Register = 00000(0)

Write Back Stage:

Inputs:

WB Controls: RegWrite= 0

MemReg= 0 ALU Result= 0000000000000101(5)

Read Data (from memory): 0000000000000000(0)

Write Register = 01010(10)

Outputs:

RegFile Before Writing: [0000000000000000, 0000000000000100, 0000000000000101, 0000000000000111, 0000000000001111, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000001100, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

RegFile After Writing: [0000000000000000, 0000000000000100, 0000000000000101, 0000000000000111, 0000000000001111, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000001100, 0000000000000000, 0000000000000101, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000, 0000000000000000]

~~~ Cycle 18 Ended ~~~

Register File:

$0: 0000000000000000(0) $1: 0000000000000100(4) $2: 0000000000000101(5) $3: 0000000000000111(7)

$4: 0000000000001111(15) $5: 0000000000000000(0) $6: 0000000000000101(5) $7: 0000000000000000(0)

$8: 0000000000001100(12) $9: 0000000000000000(0) $10: 0000000000000101(5) $11: 0000000000000000(0)

$12: 0000000000000000(0) $13: 0000000000000000(0) $14: 0000000000000000(0) $15: 0000000000000000(0)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

[0: 9]= 0, 0, 0, 0, 5, 0, 0, 0, 0, 0,

[10: 19]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[20: 29]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[30: 39]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[40: 49]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[50: 59]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[60: 69]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[70: 79]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[80: 89]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[90: 99]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[100: 109]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[110: 119]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[120: 129]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[130: 139]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[140: 149]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[150: 159]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[160: 169]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[170: 179]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[180: 189]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[190: 199]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[200: 209]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[210: 219]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[220: 229]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[230: 239]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[240: 249]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[250: 259]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[260: 269]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[270: 279]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[280: 289]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[290: 299]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[300: 309]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[310: 319]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[320: 329]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[330: 339]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[340: 349]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[350: 359]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[360: 369]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[370: 379]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[380: 389]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[390: 399]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[400: 409]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[410: 419]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[420: 429]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[430: 439]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[440: 449]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[450: 459]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[460: 469]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[470: 479]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[480: 489]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[490: 499]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[500: 509]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[510: 519]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[520: 529]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[530: 539]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[540: 549]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[550: 559]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[560: 569]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[570: 579]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[580: 589]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[590: 599]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[600: 609]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[610: 619]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[620: 629]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[630: 639]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[640: 649]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[650: 659]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[660: 669]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[670: 679]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[680: 689]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[690: 699]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[700: 709]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[710: 719]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[720: 729]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[730: 739]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[740: 749]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[750: 759]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[760: 769]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[770: 779]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[780: 789]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[790: 799]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[800: 809]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[810: 819]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[820: 829]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[830: 839]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[840: 849]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[850: 859]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[860: 869]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[870: 879]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[880: 889]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[890: 899]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[900: 909]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[910: 919]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[920: 929]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[930: 939]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[940: 949]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[950: 959]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[960: 969]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[970: 979]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[980: 989]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[990: 999]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[1000: 1009]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[1010: 1019]= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

[1020: 1029]= 0, 0, 0, 0,