Faculty of Computer & Information Sciences
Ain Shams University
Subject: CHW-362Computer Architecture & Org.
Instructors: Dr. Heba Khaled & Dr. Karim Emara
Year: 3rd year undergraduate
Academic year: 2nd term 2019-2020

Practical Project

MIPS Pipelined Emulator – Version (3)

Task2 Report:

1. Brief description of the implementation and how the code is organized.

The code written with C#, We have class called **emulator** that have the mips emulator attribute and methods and the form gui.

1.1. The class data member:

1.1.1. the 32 Mips registers `int[] mips_reg`

1.1.2. the Instruction hash set that hold the pc as key and instruction binary code as value `Dictionary<uint, string> instruction_set`

1.1.3. The program counter `uint PC`

1.1.4. The 4 pipeline registers `if_id_reg, id_ex_reg, ex_mem_reg, mem_wb_reg` implemented as structs that hold the data in each stage or cycle.

1.1.5. The 5 queues that handle cycle method `Queue<uint> f_Q, d_Q, ex_Q, mem_Q, wb_Q`

1.2. The class methods:

1.2.1. The constructor that initialize all data member.

1.2.2. The class have 5 private regions for 5 stages:

1.2.2.1. Fitch region: Only have fitch method that get the instruction of the PC then increase it by 4.

1.2.2.2. Decode region: Have decode method that divide the instruction to opcode and rs, rt and in case of r-type(rd, shamt, funct) otherwise to 16bit address and fill the `id_ex_reg` register, then take control of decode region methods:

1.2.2.2.1. control unit method that assign the control signal in the pipeline registers.

1.2.2.2.2. register file that assign the rs and rt of the instruction to `id_ex_reg` register.

1.2.2.2.3. sign extension method that extend 16bit address to 32bit.

1.2.2.3. Execute region: Have Execute method that fill the `ex_mem_reg` register then take control of execute region methods:

1.2.2.3.1. Shift left by two method

1.2.2.3.2. Address adder method that generate the branched address in case of i-type.

1.2.2.3.3. ALUSrc MUX method

1.2.2.3.4. ALU method that generate the method.

1.2.2.3.5. RIGDst MUX method

1.2.2.4. Memory access region: Have memoryAccess method that fill the `mem_wb_reg` register then take control of memory access region methods:

1.2.2.4.1. PCSrc MUX method

1.2.2.4.2. Data memory method

1.2.2.5. Write back region: Have writeBack method take control of memory access region methods:

1.2.2.5.1. MEM-to-REG MUX method

1.2.2.5.2. Register file overwrite method that write back the result to the register file.

1.2.3. The controller of the five stages the public cycle method

Use the pre implemented five queues to handle pipeline concept.

2. High-level pseudo-code for “Run 1 cycle” function.

In the emulator class:

```
 Cycle()
If wb_queue not empty then
    writeback()
    wb_queue.Dequeue()
If mem_queue not empty then
    memoryAccess()
    mem_queue.Dequeue()
If ex_queue not empty then
    execute()
    ex_queue.Dequeue()
If d_queue not empty then
    decode()
    d_queue.Dequeue()
If f_queue not empty then
    fitch()
    f_queue.Dequeue()
End Cycle
```

Then in the GUI class:

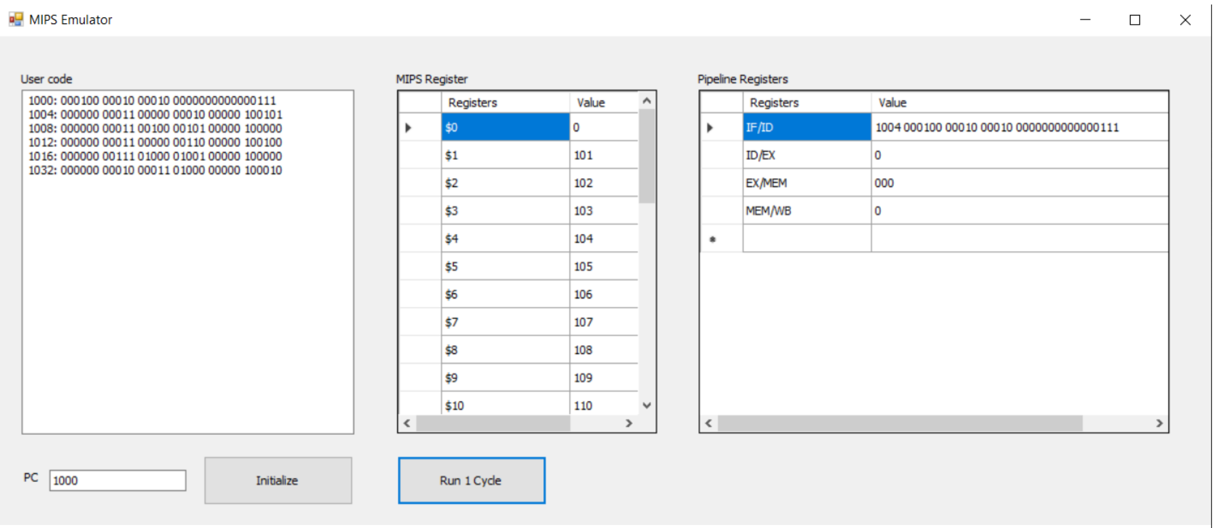
```
 runCycleBTN_Click()
 MIPS_emulator.Cycle()
 Fill_Mips_DGV()
 Fill_pipeline_DGV()
 End runCycleBTN_Click
```

3. A user guide explaining how to use the emulator

It is very simple at first the user enters the instruction binary code in user code text box
Then write the program counter that he wants to start with.
Then click initialize button this well initialize the value of mips and pipeline registers data grid view.
Then click Run 1 cycle button to for generating cycle every time you click this button the mips emulator generate one cycle.

4. Screenshots of emulating the MIPS code given in Task 1, one screenshot per clock cycle for 9 clock cycles

```
1000: 000100 00010 00010 0000000000000111
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1032: 000000 00010 00011 01000 00000 100010
```



User code

```
1000: 000100 00010 00010 0000000000000111
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1032: 000000 00010 00011 01000 00000 100010
```

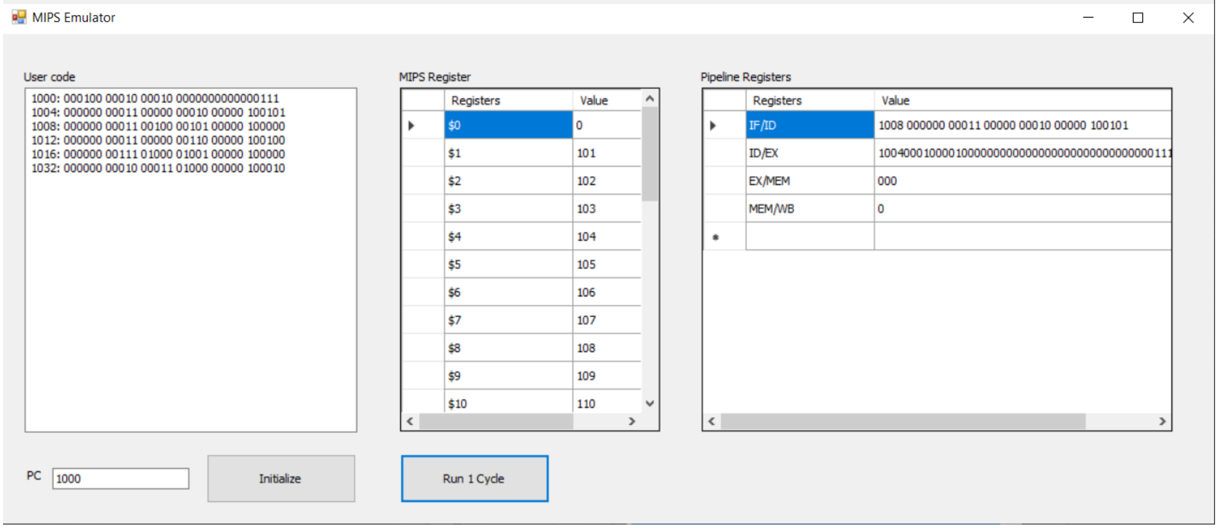
MIPS Register

Registers	Value
$0	0
$1	101
$2	102
$3	103
$4	104
$5	105
$6	106
$7	107
$8	108
$9	109
$10	110

Pipeline Registers

Registers	Value
IF/ID	1004 000100 00010 00010 0000000000000111
ID/EX	0
EX/MEM	000
MEM/WB	0

PC 1000



User code

```
1000: 000100 00010 00010 0000000000000111
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1032: 000000 00010 00011 01000 00000 100010
```

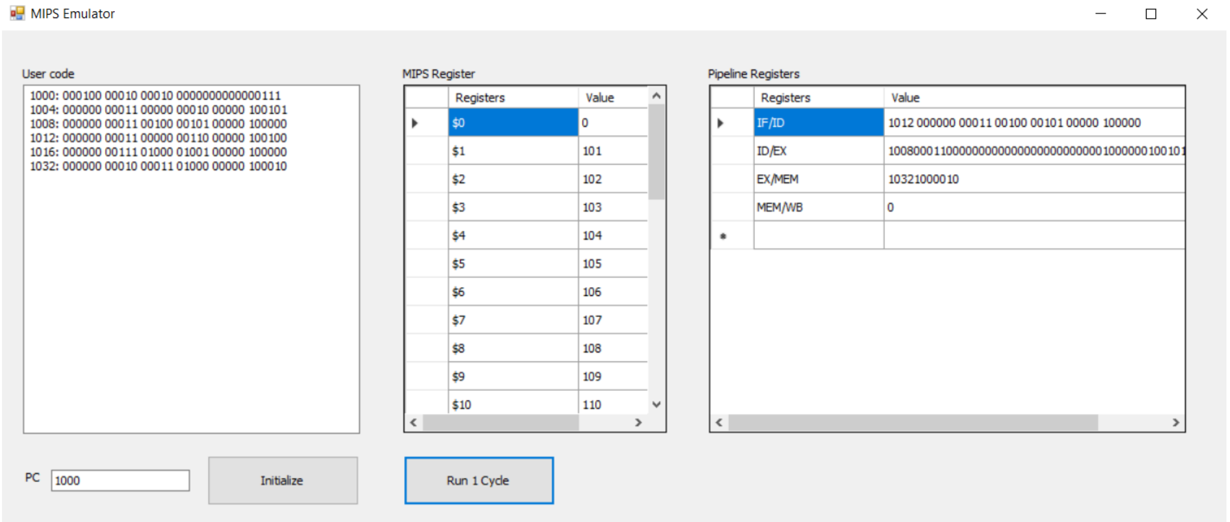
MIPS Register

Registers	Value
$0	0
$1	101
$2	102
$3	103
$4	104
$5	105
$6	106
$7	107
$8	108
$9	109
$10	110

Pipeline Registers

Registers	Value
IF/ID	1008 000000 00011 00000 00010 00000 100101
ID/EX	10040001000010000000000000000000000000000111
EX/MEM	000
MEM/WB	0

PC 1000



User code

```
1000: 000100 00010 00010 0000000000000111
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1032: 000000 00010 00011 01000 00000 100010
```

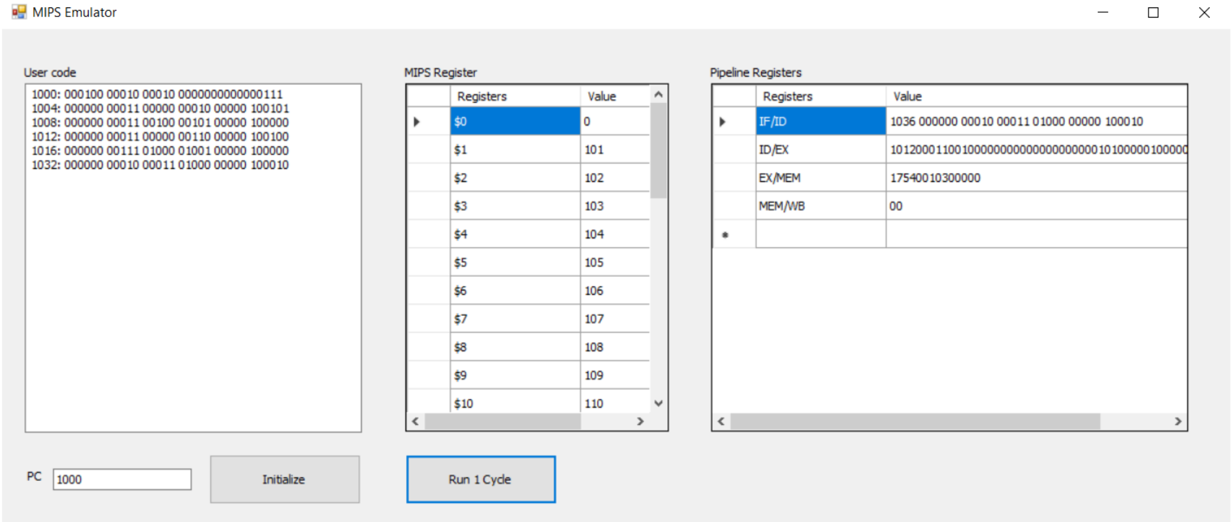
MIPS Register

Registers	Value
$0	0
$1	101
$2	102
$3	103
$4	104
$5	105
$6	106
$7	107
$8	108
$9	109
$10	110

Pipeline Registers

Registers	Value
IF/ID	1012 000000 00011 00100 00101 00000 100000
ID/EX	10080001100000000000000000000000001000000100101
EX/MEM	1032100010
MEM/WB	0

PC 1000



User code

```
1000: 000100 00010 00010 0000000000000111
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1032: 000000 00010 00011 01000 00000 100010
```

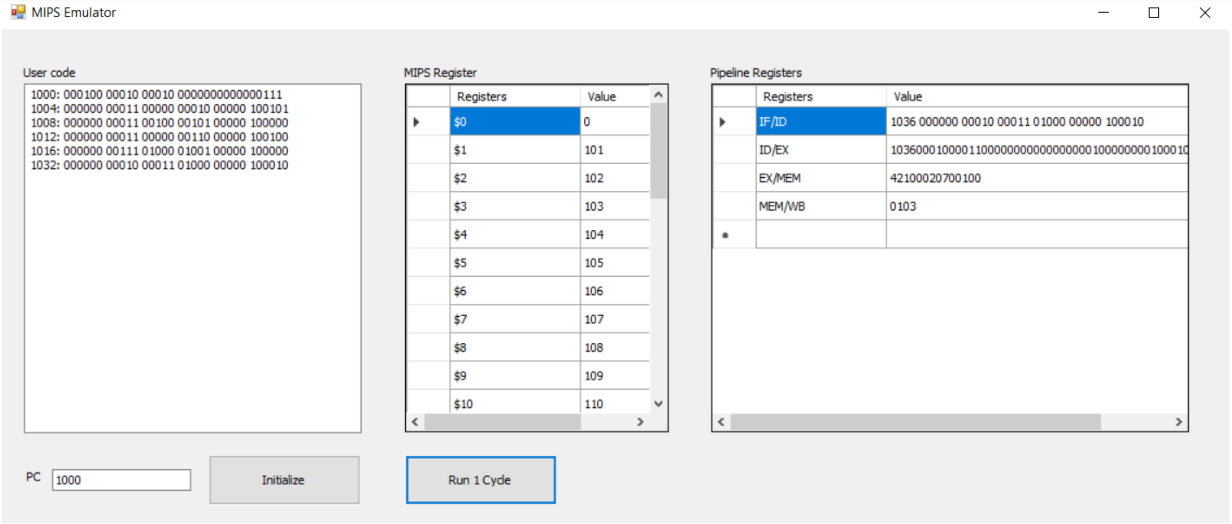
MIPS Register

Registers	Value
$0	0
$1	101
$2	102
$3	103
$4	104
$5	105
$6	106
$7	107
$8	108
$9	109
$10	110

Pipeline Registers

Registers	Value
IF/ID	1036 000000 00010 00011 01000 00000 100010
ID/EX	1012000110010000000000000000000001010000010000
EX/MEM	1754001030000
MEM/WB	00

PC 1000



User code

```
1000: 000100 00010 00010 0000000000000111
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1032: 000000 00010 00011 01000 00000 100010
```

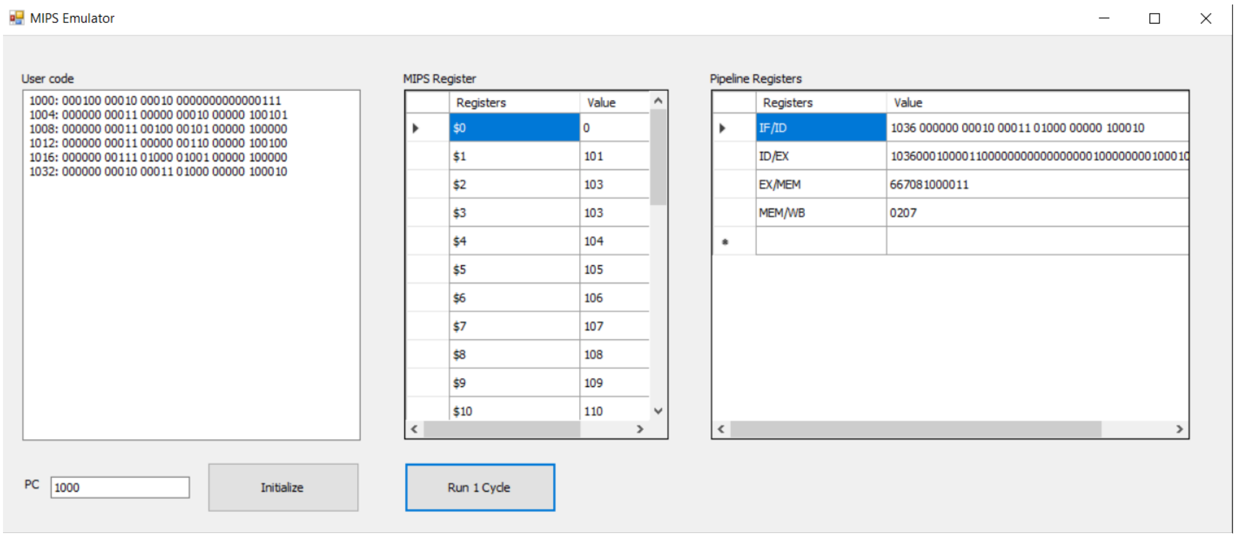
MIPS Register

Registers	Value
$0	0
$1	101
$2	102
$3	103
$4	104
$5	105
$6	106
$7	107
$8	108
$9	109
$10	110

Pipeline Registers

Registers	Value
IF/ID	1036 000000 00010 00011 01000 00000 100010
ID/EX	1036000100001000000000000000000010000000010001 0
EX/MEM	42100020700100
MEM/WB	0103

PC 1000



User code

```
1000: 000100 00010 00010 0000000000000111
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1032: 000000 00010 00011 01000 00000 100010
```

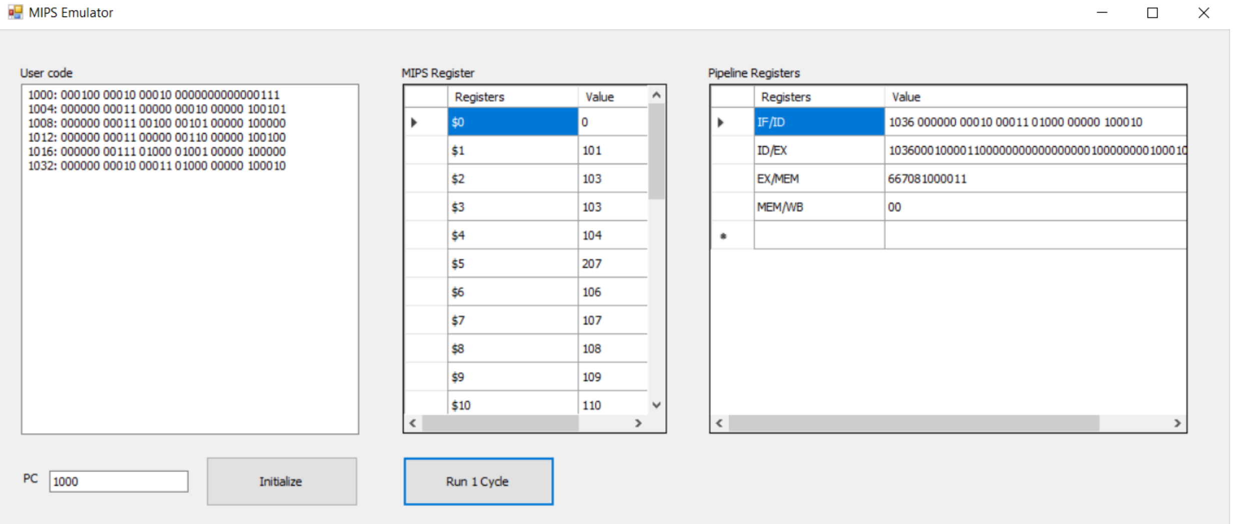
MIPS Register

Registers	Value
$0	0
$1	101
$2	103
$3	103
$4	104
$5	105
$6	106
$7	107
$8	108
$9	109
$10	110

Pipeline Registers

Registers	Value
IF/ID	1036 000000 00010 00011 01000 00000 100010
ID/EX	1036000100001100000000000000000100000000100010
EX/MEM	667081000011
MEM/WB	0207

PC 1000



User code

```
1000: 000100 00010 00010 0000000000000111
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1032: 000000 00010 00011 01000 00000 100010
```

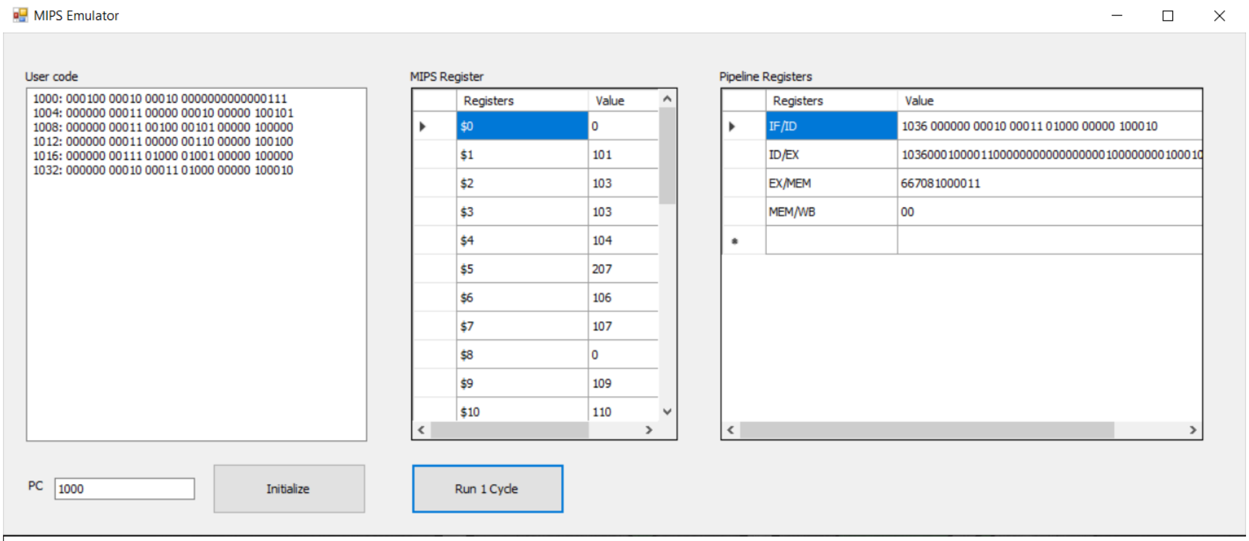
MIPS Register

Registers	Value
$0	0
$1	101
$2	103
$3	103
$4	104
$5	207
$6	106
$7	107
$8	108
$9	109
$10	110

Pipeline Registers

Registers	Value
IF/ID	1036 000000 00010 00011 01000 00000 100010
ID/EX	1036000100001100000000000000000100000000100010
EX/MEM	667081000011
MEM/WB	00

PC 1000



User code

```
1000: 000100 00010 00010 0000000000000111
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1032: 000000 00010 00011 01000 00000 100010
```

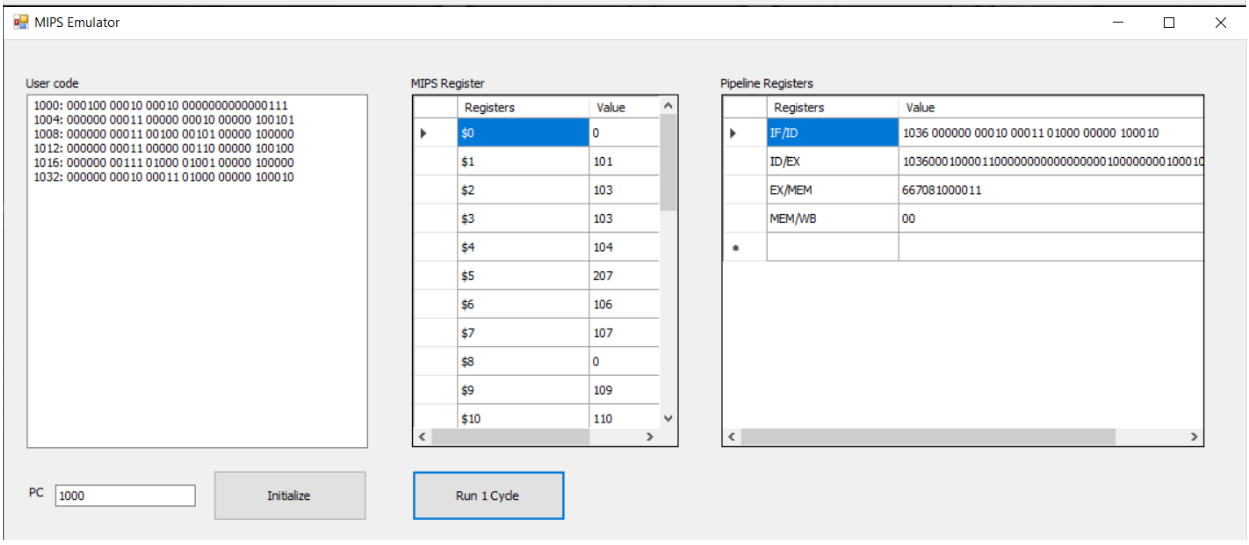
MIPS Register

	Registers	Value
▶	$0	0
	$1	101
	$2	103
	$3	103
	$4	104
	$5	207
	$6	106
	$7	107
	$8	0
	$9	109
	$10	110

Pipeline Registers

	Registers	Value
▶	IF/ID	1036 000000 00010 00011 01000 00000 100010
	ID/EX	1036000100001100000000000000000000100000000100010
	EX/MEM	667081000011
	MEM/WB	00
*		

PC 1000



User code

```
1000: 000100 00010 00010 0000000000000111
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1032: 000000 00010 00011 01000 00000 100010
```

MIPS Register

	Registers	Value
▶	$0	0
	$1	101
	$2	103
	$3	103
	$4	104
	$5	207
	$6	106
	$7	107
	$8	0
	$9	109
	$10	110

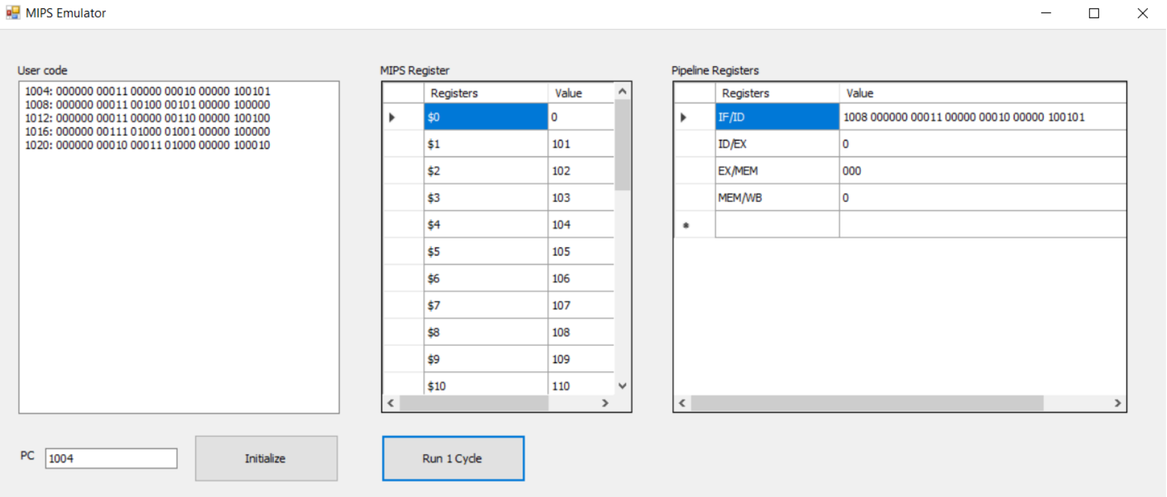
Pipeline Registers

	Registers	Value
▶	IF/ID	1036 000000 00010 00011 01000 00000 100010
	ID/EX	1036000100001100000000000000000000100000000100010
	EX/MEM	667081000011
	MEM/WB	00
*		

PC 1000

5. Screenshots of emulating another MIPS code from your choice, one screenshot per clock cycle for 9 clock cycles

```
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1020: 000000 00010 00011 01000 00000 100010
```



MIPS Emulator

User code

```
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1020: 000000 00010 00011 01000 00000 100010
```

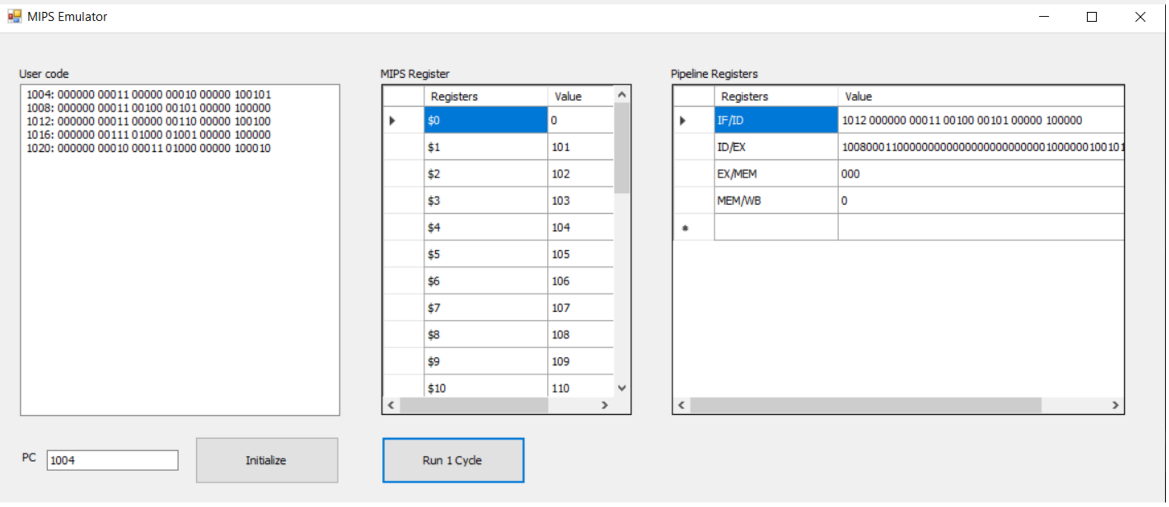
MIPS Register

Registers	Value
$0	0
$1	101
$2	102
$3	103
$4	104
$5	105
$6	106
$7	107
$8	108
$9	109
$10	110

Pipeline Registers

Registers	Value
IF/ID	1008 000000 00011 00000 00010 00000 100101
ID/EX	0
EX/MEM	000
MEM/WB	0

PC 1004



MIPS Emulator

User code

```
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1020: 000000 00010 00011 01000 00000 100010
```

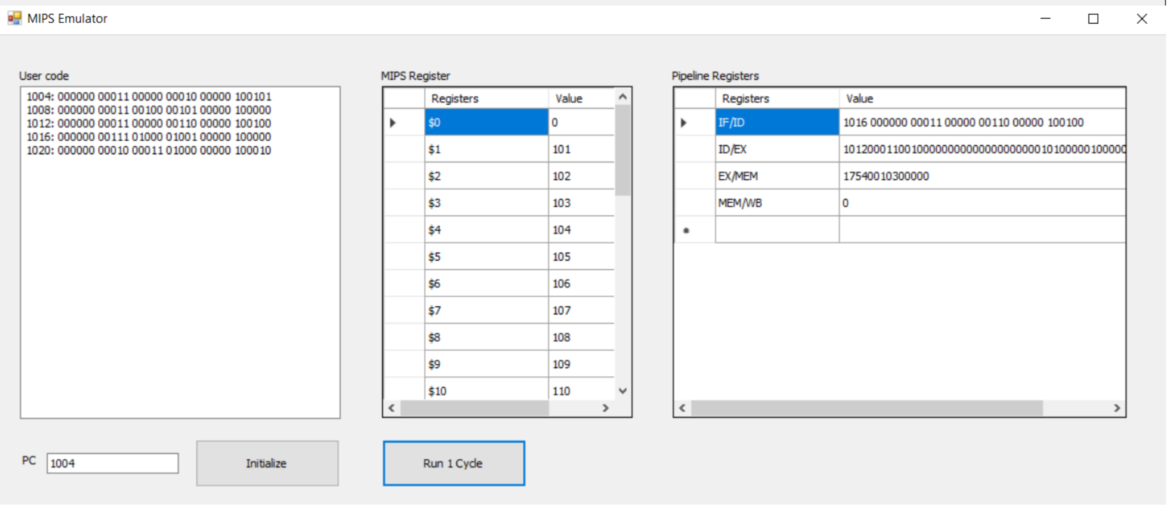
MIPS Register

Registers	Value
$0	0
$1	101
$2	102
$3	103
$4	104
$5	105
$6	106
$7	107
$8	108
$9	109
$10	110

Pipeline Registers

Registers	Value
IF/ID	1012 000000 00011 00100 00101 00000 100000
ID/EX	1008000110000000000000000000000001000000100101
EX/MEM	000
MEM/WB	0

PC 1004



MIPS Emulator

User code

```
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1020: 000000 00010 00011 01000 00000 100010
```

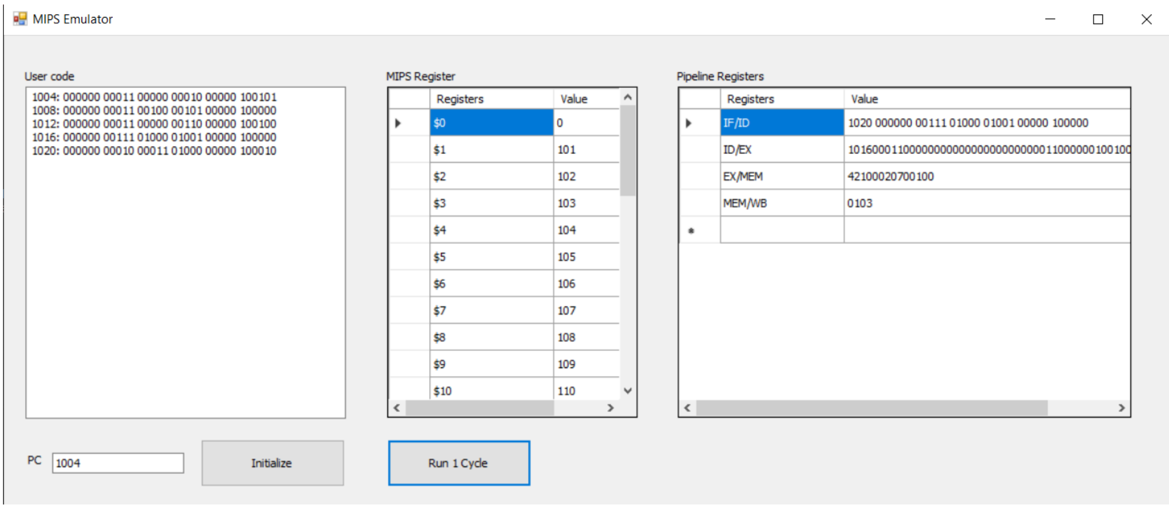
MIPS Register

Registers	Value
$0	0
$1	101
$2	102
$3	103
$4	104
$5	105
$6	106
$7	107
$8	108
$9	109
$10	110

Pipeline Registers

Registers	Value
IF/ID	1016 000000 00011 00000 00110 00000 100100
ID/EX	10120001100100000000000000000000010100000100000
EX/MEM	1754001030000
MEM/WB	0

PC 1004



User code

```
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1020: 000000 00010 00011 01000 00000 100010
```

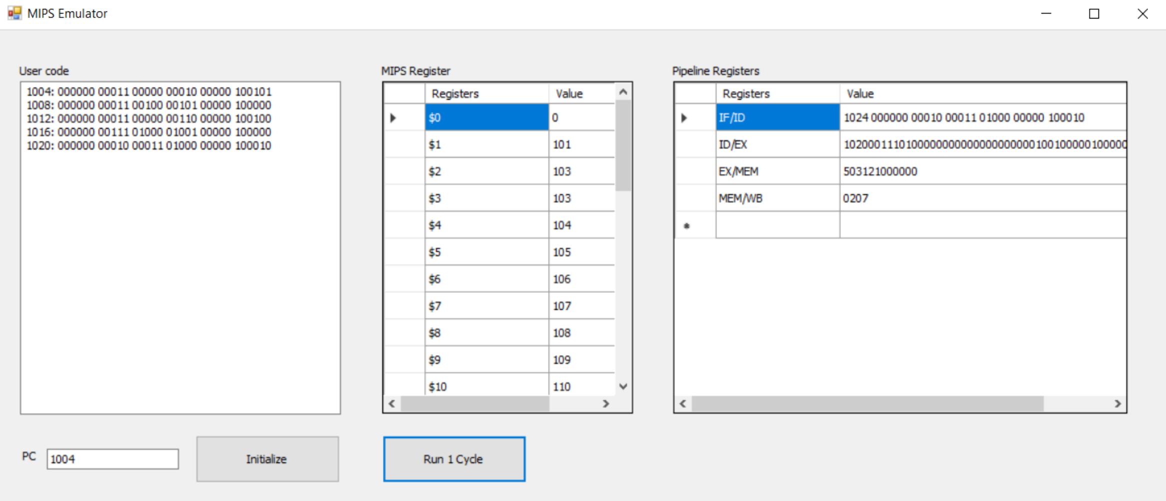
MIPS Register

Registers	Value
$0	0
$1	101
$2	102
$3	103
$4	104
$5	105
$6	106
$7	107
$8	108
$9	109
$10	110

Pipeline Registers

Registers	Value
IF/ID	1020 000000 00111 01000 01001 00000 100000
ID/EX	10160001100000000000000000000000001100000010010
EX/MEM	42100020700100
MEM/WB	0103

PC 1004



User code

```
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1020: 000000 00010 00011 01000 00000 100010
```

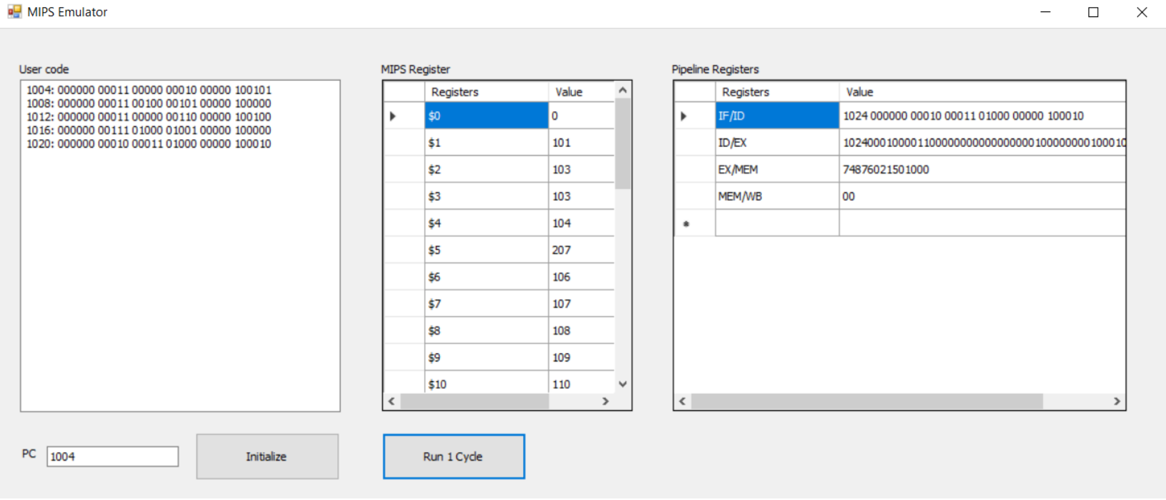
MIPS Register

Registers	Value
$0	0
$1	101
$2	103
$3	103
$4	104
$5	105
$6	106
$7	107
$8	108
$9	109
$10	110

Pipeline Registers

Registers	Value
IF/ID	1024 000000 00010 00011 01000 00000 100010
ID/EX	10200011101000000000000000000100100000100000
EX/MEM	503121000000
MEM/WB	0207

PC 1004



User code

```
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1020: 000000 00010 00011 01000 00000 100010
```

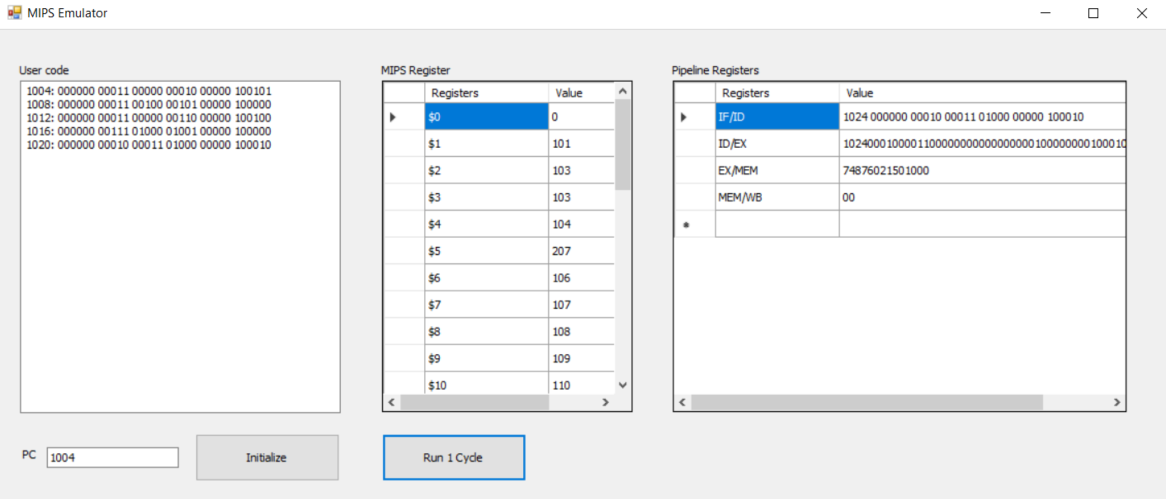
MIPS Register

Registers	Value
$0	0
$1	101
$2	103
$3	103
$4	104
$5	207
$6	106
$7	107
$8	108
$9	109
$10	110

Pipeline Registers

Registers	Value
IF/ID	1024 000000 00010 00011 01000 00000 100010
ID/EX	1024000100001100000000000000000010000000010001
EX/MEM	74876021501000
MEM/WB	00

PC 1004



User code

```
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1020: 000000 00010 00011 01000 00000 100010
```

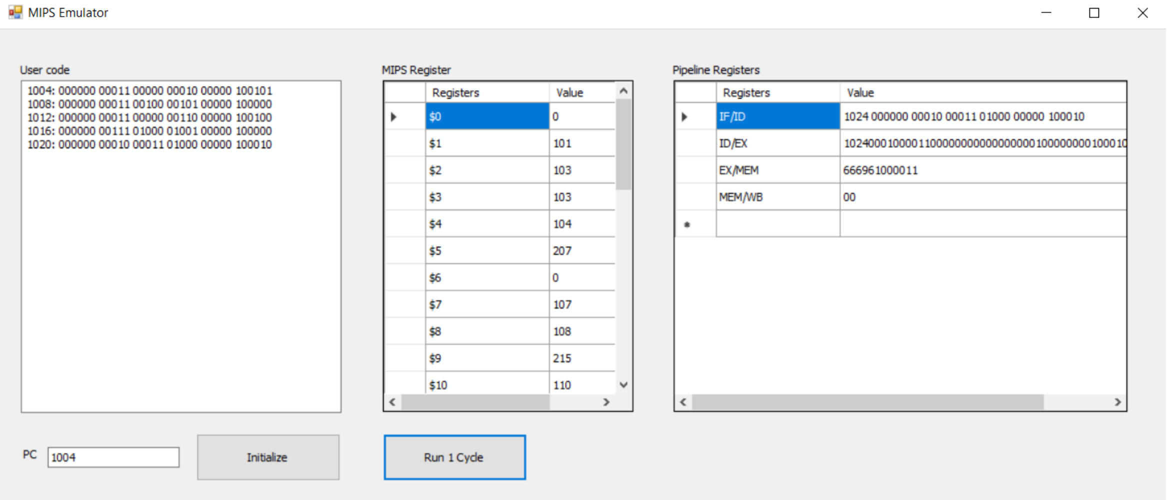
MIPS Register

Registers	Value
$0	0
$1	101
$2	103
$3	103
$4	104
$5	207
$6	106
$7	107
$8	108
$9	109
$10	110

Pipeline Registers

Registers	Value
IF/ID	1024 000000 00010 00011 01000 00000 100010
ID/EX	10240001000011000000000000000000100000000100010
EX/MEM	74876021501000
MEM/WB	00

PC 1004



User code

```
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1020: 000000 00010 00011 01000 00000 100010
```

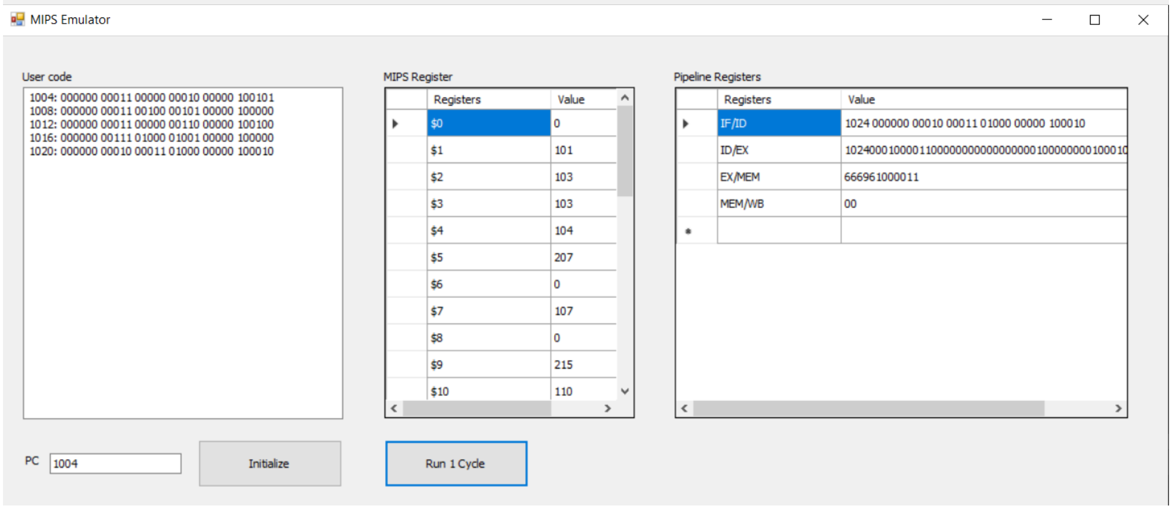
MIPS Register

Registers	Value
$0	0
$1	101
$2	103
$3	103
$4	104
$5	207
$6	0
$7	107
$8	108
$9	215
$10	110

Pipeline Registers

Registers	Value
IF/ID	1024 000000 00010 00011 01000 00000 100010
ID/EX	10240001000011000000000000000000100000000100010
EX/MEM	666961000011
MEM/WB	00

PC 1004



User code

```
1004: 000000 00011 00000 00010 00000 100101
1008: 000000 00011 00100 00101 00000 100000
1012: 000000 00011 00000 00110 00000 100100
1016: 000000 00111 01000 01001 00000 100000
1020: 000000 00010 00011 01000 00000 100010
```

MIPS Register

Registers	Value
$0	0
$1	101
$2	103
$3	103
$4	104
$5	207
$6	0
$7	107
$8	0
$9	215
$10	110

Pipeline Registers

Registers	Value
IF/ID	1024 000000 00010 00011 01000 00000 100010
ID/EX	10240001000011000000000000000000100000000100010
EX/MEM	666961000011
MEM/WB	00

PC 1004