

# ROTEIRO 8

Stage processor w/o forwarding unit:

## Código 1:

Código de máquina armazenado na memória

The screenshot shows the Ripes IDE interface. On the left, there's a sidebar with icons for Processor, Cache, Memory, and I/O. The main window is divided into three panes: Source code, Disassembled code, and GPR.

**Source code:**

```
1 addi x10, x0, 4
2 addi x11, x0, 3
3 addi x12, x0, 7
4 addi x13, x0, 5
5 addi x14, x0, 6
6 add x15, x10, x11
7
```

**Disassembled code:**

```
0: 00400513 addi x10 x0 4 IF
4: 00300593 addi x11 x0 3
8: 00700613 addi x12 x0 7
c: 00500693 addi x13 x0 5
10: 00600713 addi x14 x0 6
14: 00b507b3 add x15 x10 x11
```

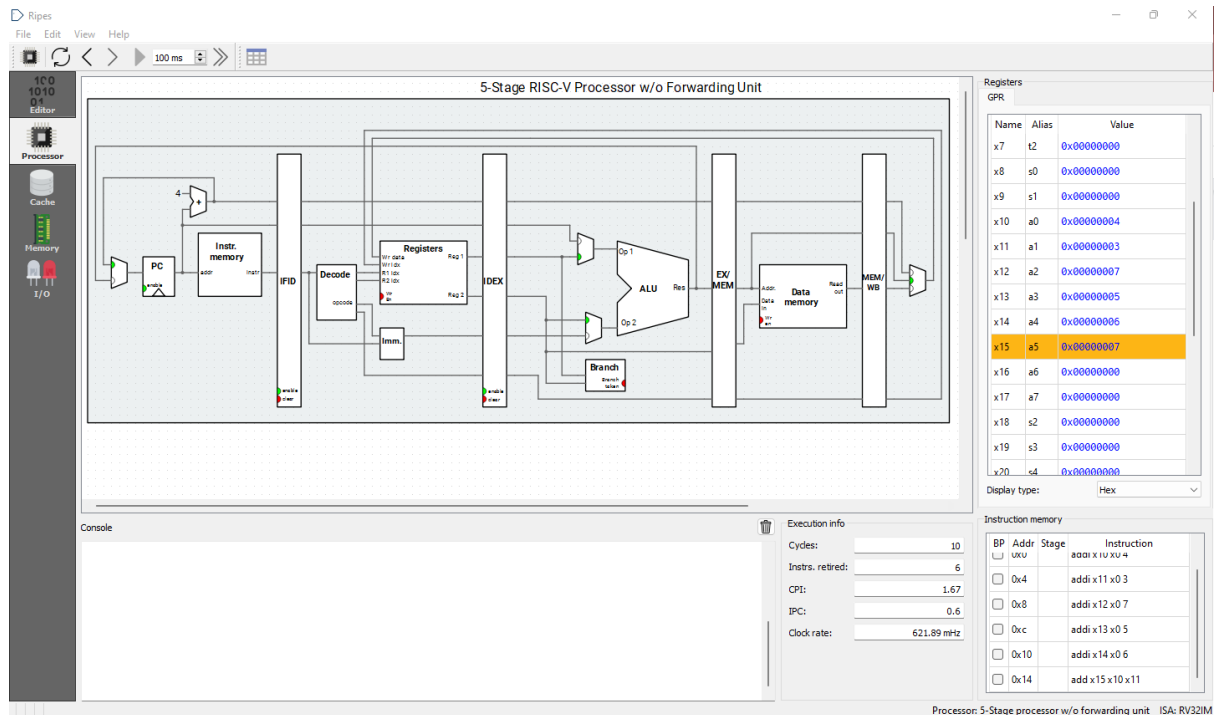
**GPR:**

Name	Alias	Value
x0	zero	0x00000000
x1	ra	0x00000000
x2	sp	0x7ffffff0
x3	gp	0x10000000
x4	tp	0x00000000
x5	t0	0x00000000
x6	t1	0x00000000
x7	t2	0x00000000
x8	s0	0x00000000
x9	s1	0x00000000
x10	a0	0x00000000
x11	a1	0x00000000
x12	a2	0x00000000
x13	a3	0x00000000
x14	a4	0x00000000
x15	a5	0x00000000
x16	a6	0x00000000
x17	a7	0x00000000
x18	s2	0x00000000
x19	s3	0x00000000
x20	s4	0x00000000
x21	s5	0x00000000

Display type: Hex

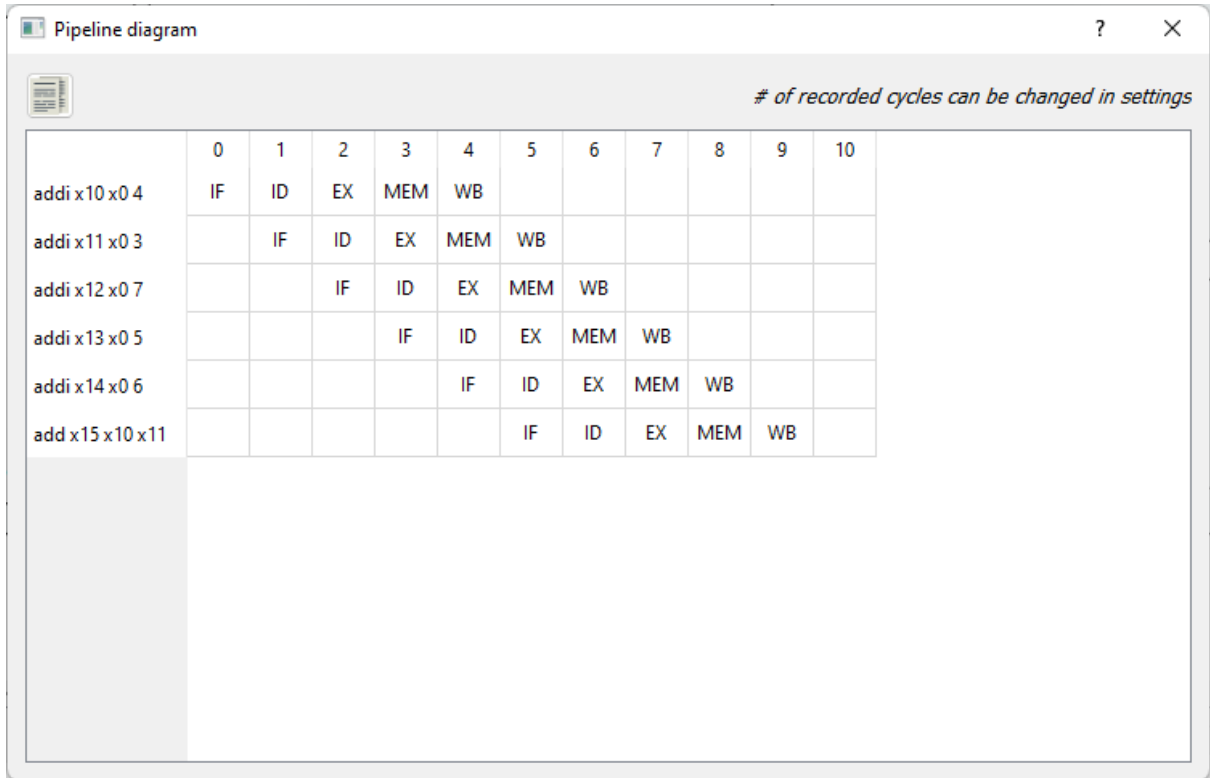
Processor: 5-Stage processor w/o forwarding unit ISA: RV32IM

Após o fim da execução do código:



- **Informações de execução:**  
Houveram 10 ciclos pois o código continha 6 instruções, onde a única instrução que tinha dependência com anteriores, quando executada, tinha suas dependências totalmente executadas, fazendo possível o melhor proveito do pipeline com o número de ciclos determinado por  $1 * e + (n - 1) * 1$ , sendo nesta situação  $e$  número de estágio e  $n$  número de instruções iguais, ambos, a 5.
- **Registradores:**
  - Em x10 foi salvo o valor 4;
  - Em x11 foi salvo o valor 3;
  - Em x12 foi salvo o valor 7;
  - Em x13 foi salvo o valor 5;
  - Em x14 foi salvo o valor 6;
  - Em x15 foi salvo o valor da soma dos registradores x10 e x11;

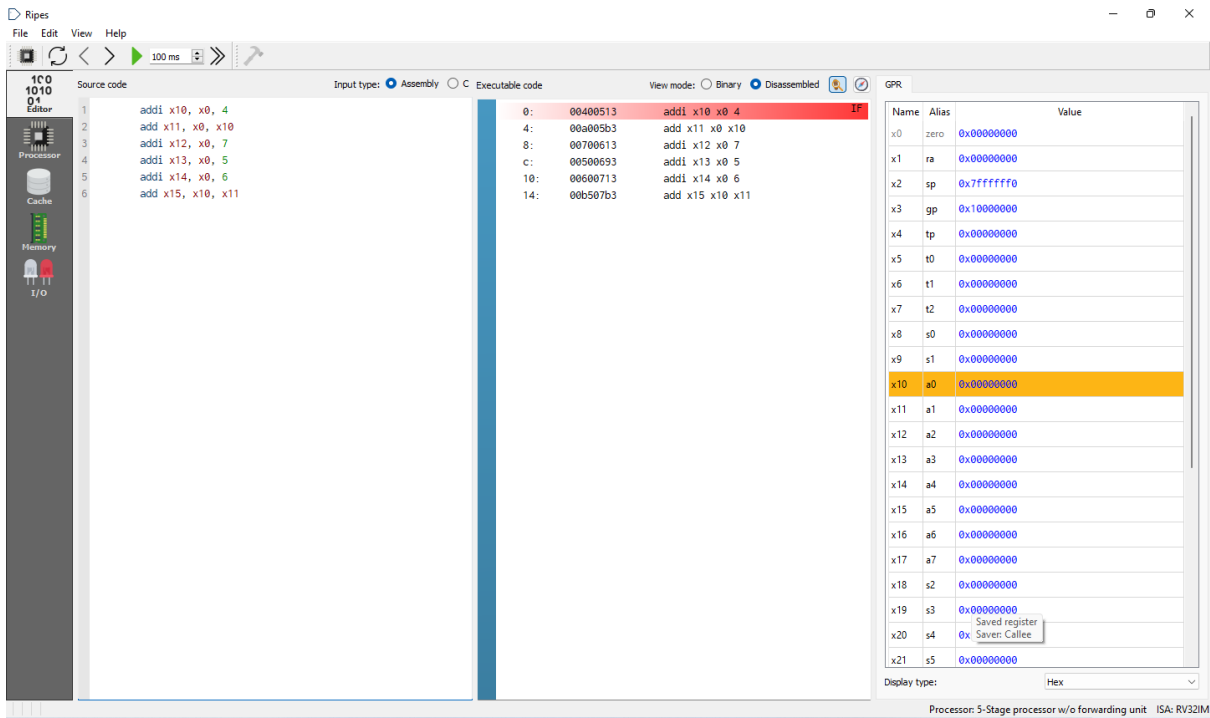
Diagrama de pipeline:



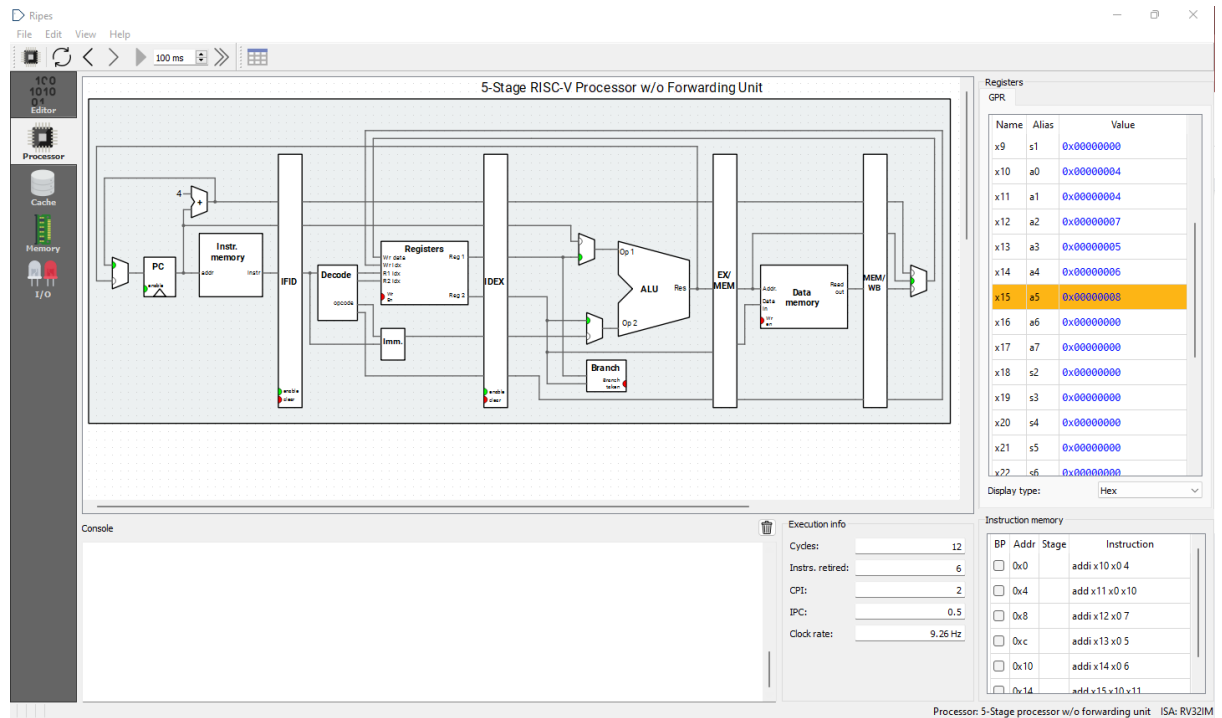
Fluxo normal sem travas por dependência

Código 2:

Código de máquina armazenado na memória

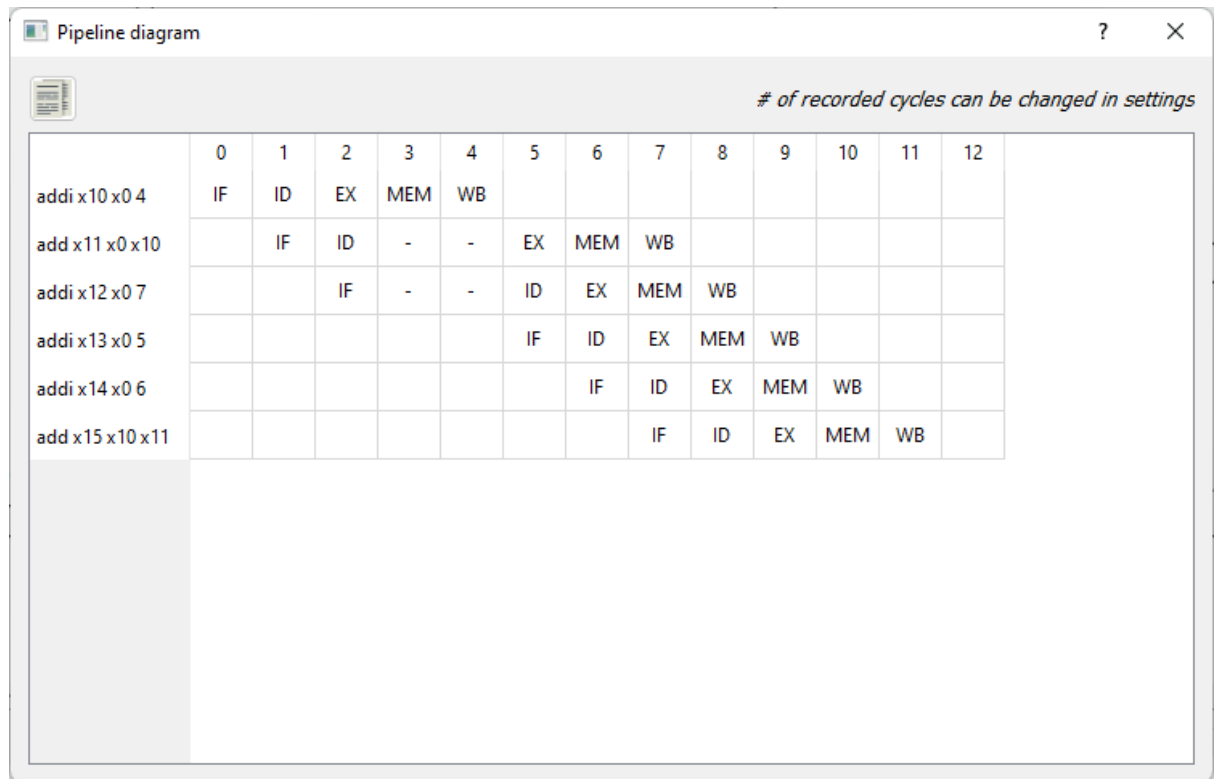


Após o fim da execução do código:



- **Informações de execução:**  
Houveram 12 ciclos pois o código continha 6 instruções, onde já na segunda instrução havia uma dependência fazendo com que houvesse uma espera de dois ciclos para que a informação dependida fosse finalizada para ser usada. Na última instrução também havia dependência, mas que, quando executada, tinha suas dependências totalmente executadas. tendo assim  $1 * e + (n - 1) * 1 + 2$ , sendo nesta situação  $e$  número de estágios e  $n$  número de instruções iguais, ambos, a 5.
- **Registradores:**
  - Em x10 foi salvo o valor 4;
  - Em x11 foi salvo o valor contido no registrador x10;
  - Em x12 foi salvo o valor 7;
  - Em x13 foi salvo o valor 5;
  - Em x14 foi salvo o valor 6;
  - Em x15 foi salvo o valor da soma dos registradores x10 e x11;

Diagrama de pipeline:



Fluxo normal até o segundo ciclo, onde até o ciclo 4 há a espera para a finalização de acesso a memória da instrução 1 que contém registrador que é usado na instrução 2, travando a execução do código com a instrução 2 parada no ciclo de identificação de instrução e a instrução 1 na busca de instrução.

Stage processor w/o forwarding or hazard detection:

## Código 1:

Código de máquina armazenado na memória

The screenshot shows the Ripes IDE interface. On the left, there's a sidebar with icons for Editor, Processor, Cache, Memory, and I/O. The main window is divided into three panes. The left pane shows the source code in assembly:

```
1 addi x10, x0, 4
2 addi x11, x0, 3
3 addi x12, x0, 7
4 addi x13, x0, 5
5 addi x14, x0, 6
6 add x15, x10, x11
```

The middle pane shows the disassembled instructions with their binary and decimal representations:

Address	Hex	Assembly
0:	00400513	addi x10 x0 4
4:	00300593	addi x11 x0 3
8:	00700613	addi x12 x0 7
c:	00500693	addi x13 x0 5
10:	00600713	addi x14 x0 6
14:	00b507b3	add x15 x10 x11

The right pane shows the GPR (General Purpose Register) table:

Name	Alias	Value
x0	zero	0x00000000
x1	ra	0x00000000
x2	sp	0x7fffffff0
x3	gp	0x10000000
x4	tp	0x00000000
x5	t0	0x00000000
x6	t1	0x00000000
x7	t2	0x00000000
x8	s0	0x00000000
x9	s1	0x00000000
x10	a0	0x00000000
x11	a1	0x00000000
x12	a2	0x00000000
x13	a3	0x00000000
x14	a4	0x00000000
x15	a5	0x00000000
x16	a6	0x00000000
x17	a7	0x00000000
x18	s2	0x00000000
x19	s3	0x00000000
x20	s4	0x00000000
x21	s5	0x00000000

At the bottom, it says "Processor: 5-stage processor w/o forwarding or hazard detection ISA: RV32IM".

Após o fim da execução do código:

The screenshot shows the Ripes IDE interface after execution. The main window displays a block diagram of the "5-Stage RISC-V Processor w/o Forwarding or Hazard Detection". The diagram shows the flow of data through the stages: PC, Instruction Memory, IFID, Decode, Registers, IDEX, ALU, EX/MEM, Data Memory, and NEW/WB. The right pane shows the GPR table with updated values:

Name	Alias	Value
x7	t2	0x00000000
x8	s0	0x00000000
x9	s1	0x00000000
x10	a0	0x00000004
x11	a1	0x00000003
x12	a2	0x00000007
x13	a3	0x00000005
x14	a4	0x00000006
x15	a5	0x00000007
x16	a6	0x00000000
x17	a7	0x00000000
x18	s2	0x00000000
x19	s3	0x00000000
x20	s4	0x00000000

The bottom pane shows execution statistics:

Execution Info	Value
Cycles	10
Instrs. retired	6
CPI	1.67
IPC	0.6
Clock rate	8.93 Hz

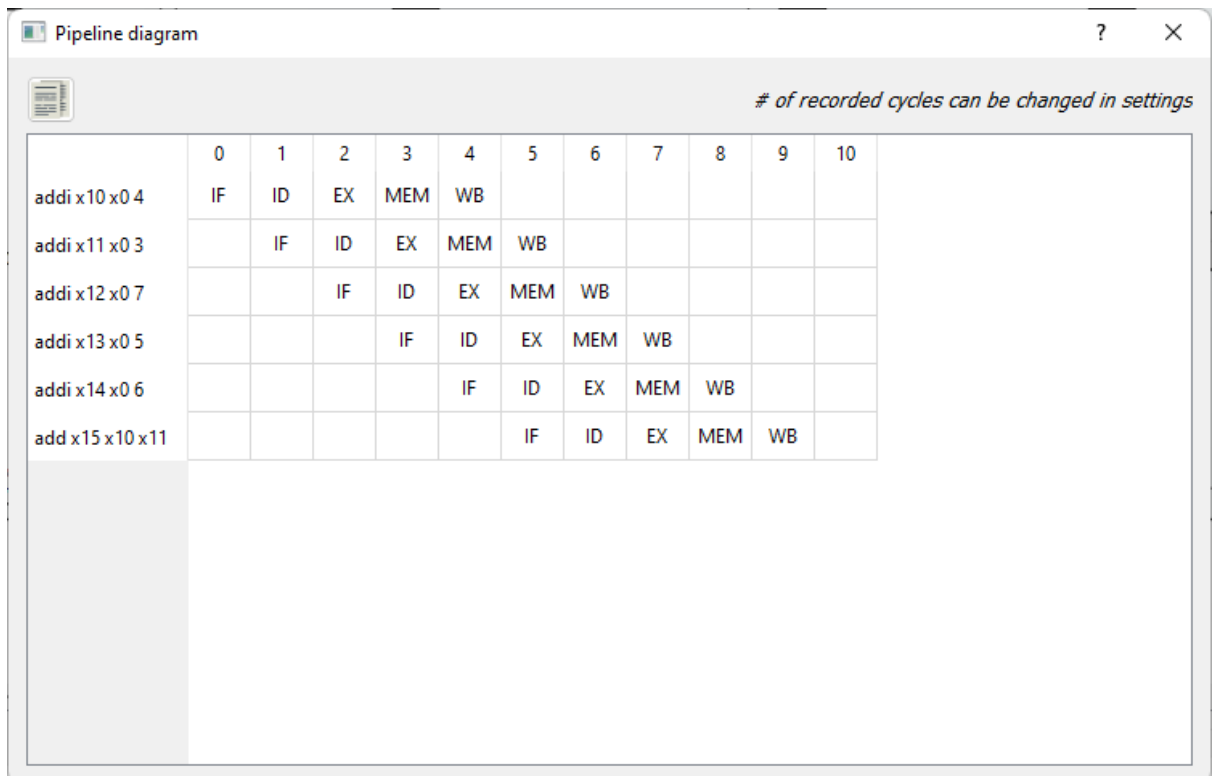
At the bottom, it says "Processor: 5-stage processor w/o forwarding or hazard detection ISA: RV32IM".

- Informações de execução:

Também houveram 10 ciclos pois o código no modo anterior já não ficava esperando a finalização de dependências.

- Registradores:
  - Em x10 foi salvo o valor 4;
  - Em x11 foi salvo o valor 3;
  - Em x12 foi salvo o valor 7;
  - Em x13 foi salvo o valor 5;
  - Em x14 foi salvo o valor 6;
  - Em x15 foi salvo o valor da soma dos registradores x10 e x11;

Diagrama de pipeline:



Fluxo normal também.

## Código 2:

### Código de máquina armazenado na memória

The screenshot shows the Ripes IDE interface. On the left, there's a sidebar with icons for Processor, Cache, Memory, and I/O. The main window is divided into three panes: Source code, Disassembled code, and GPR (General Purpose Register) table.

**Source code:**

```
1 addi x10, x0, 4
2 add x11, x0, x10
3 addi x12, x0, 7
4 addi x13, x0, 5
5 addi x14, x0, 6
6 add x15, x10, x11
```

**Disassembled code:**

Address	Hex	Assembly
0:	00400513	addi x10 x0 4
4:	00a005b3	add x11 x0 x10
8:	00700613	addi x12 x0 7
c:	00500693	addi x13 x0 5
10:	00600713	addi x14 x0 6
14:	00b507b3	add x15 x10 x11

**GPR Table:**

Name	Alias	Value
x0	zero	0x00000000
x1	ra	0x00000000
x2	sp	0x7fffffff0
x3	gp	0x10000000
x4	tp	0x00000000
x5	t0	0x00000000
x6	t1	0x00000000
x7	t2	0x00000000
x8	s0	0x00000000
x9	s1	0x00000000
x10	a0	0x00000000
x11	a1	0x00000000
x12	a2	0x00000000
x13	a3	0x00000000
x14	a4	0x00000000
x15	a5	0x00000000
x16	a6	0x00000000
x17	a7	0x00000000
x18	s2	0x00000000
x19	s3	0x00000000
x20	s4	0x00000000
x21	s5	0x00000000

Display type: Hex

Processor: 5-stage processor w/o forwarding or hazard detection ISA: RV32IM

### Após o fim da execução do código:

The screenshot shows the Ripes IDE interface after execution. The main window displays a block diagram of the 5-stage RISC-V Processor. The stages are: IFID (Instruction Fetch/Decode), Decode, IDEX (Instruction Decode/Execute), ALU (Arithmetic Logic Unit), and EX/MEM (Execute/Memory Access). The diagram shows the flow of instructions and data between these stages, including the PC (Program Counter), Instruction Memory, Registers, ALU, Data Memory, and Branch unit.

**Registers Table:**

Name	Alias	Value
x7	t2	0x00000000
x8	s0	0x00000000
x9	s1	0x00000000
x10	a0	0x00000004
x11	a1	0x00000000
x12	a2	0x00000007
x13	a3	0x00000005
x14	a4	0x00000006
x15	a5	0x00000004
x16	a6	0x00000000
x17	a7	0x00000000
x18	s2	0x00000000
x19	s3	0x00000000
x20	s4	0x00000000

**Execution Info:**

Field	Value
Cycles	10
Instrs. retired	6
CPI	1.67
IPC	0.6
Clock rate	9.01 Hz

**Instruction Memory:**

BP	Addr	Stage	Instruction
<input type="checkbox"/>	0x4	addi x11 x0 4	
<input type="checkbox"/>	0x8	addi x12 x0 7	
<input type="checkbox"/>	0xc	addi x13 x0 5	
<input type="checkbox"/>	0x10	addi x14 x0 6	
<input type="checkbox"/>	0x14	add x15 x10 x11	

Display type: Hex

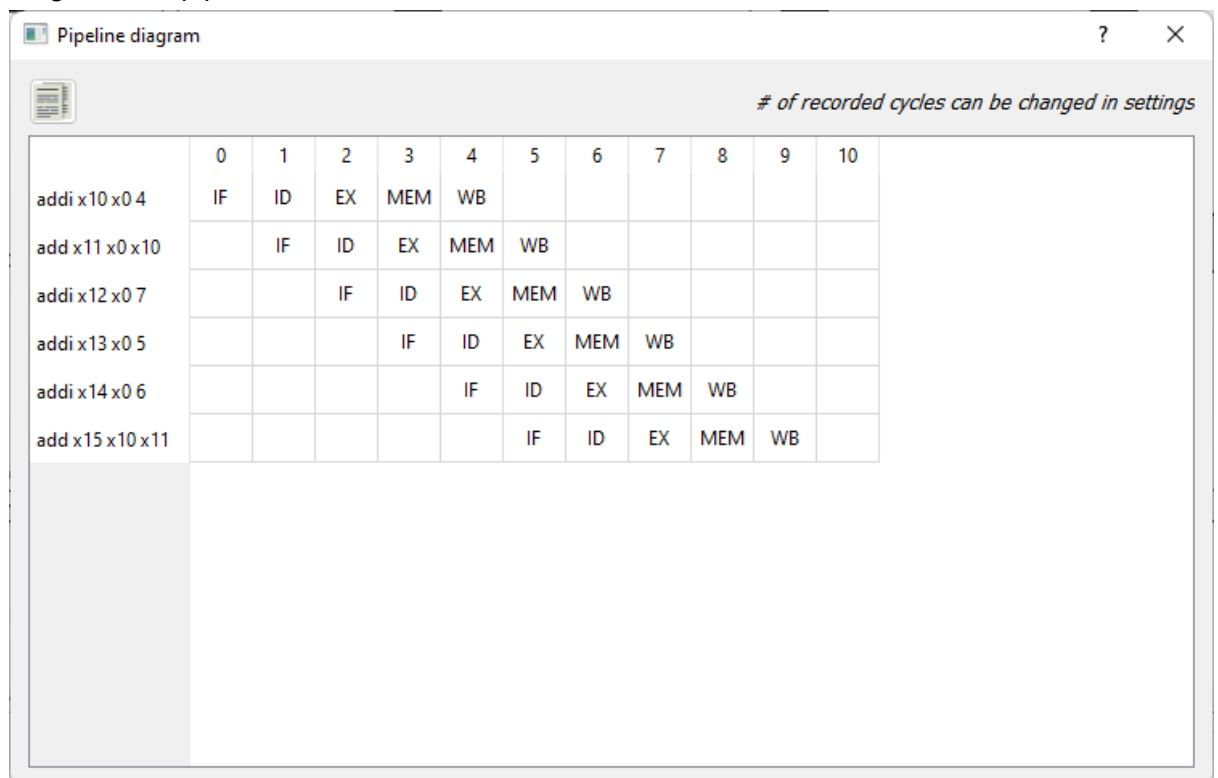
Processor: 5-stage processor w/o forwarding or hazard detection ISA: RV32IM

- Informações de execução:  
Agora houveram 10 ciclos apesar das dependências dado as otimizações desse modo de pipeline.
- Registradores:



- Em x10 foi salvo o valor 4;
- Em x11 foi salvo o valor contido no registrador x10;
- Em x12 foi salvo o valor 7;
- Em x13 foi salvo o valor 5;
- Em x14 foi salvo o valor 6;
- Em x15 foi salvo o valor da soma dos registradores x10 e x11;

Diagrama de pipeline:



Fluxo normal apesar das dependências dado as otimizações desse modo de pipeline.