

Software simulador do microcontrolador M+++

Aluno(a): Jean Carlos Klann

Orientador: Miguel Alexandre
Wisintainer

Roteiro

- 3 – Introdução
- 4 – Objetivos
- 5 – Fundamentação Teórica
- 10 – Trabalhos Correlatos
- 14 – Requisitos
- 17 – Especificação
- 21 – Implementação
- 42 – Operacionalidade da Implementação
- 44 – Resultados e Discussões
- 49 – Conclusões e Sugestões

Introdução

- Criado originalmente por Jonathan Manoel Borges;
- aprimorado por Jean Jung;
- ferramenta de auxílio-aprendizagem;
- utilizado atualmente nas aulas de Arquitetura de Computadores;

Objetivos

- Auxiliar no desenvolvimento e qualificação dos alunos;
- permitir a programação, inspeção de bugs, e uma rápida simulação do programa;
- disponibilizar um modo de depuração;
- disponibilizar uma interface de montagem de circuitos.

Fundamentação Teórica

Microcontrolador de 8 bits;

Operação ULA	Registrador / Porta de entrada ou saída	Fluxo de operação
3 bits	2 bits	3 bits

Fundamentação Teórica

Operação ULA

Binário	Operação
000	ADD
001	SUB
010	AND
011	OR
100	XOR
101	NOT
110	MOV
111	INC

Fundamentação Teórica

Registradores / portas de entrada ou saída

Código	Nome
00	B / IN0 / OUT0
01	C / IN1 / OUT1
10	D / IN2 / OUT2
11	E / IN3 / OUT3

Fundamentação Teórica

Registradores / portas de entrada ou saída

Página de código	Binário	Fluxo
00000	000	$A * A \rightarrow A$
00000	111	High decoder
00001	000	$A * ROM \rightarrow A$
00010	000	RET

Fundamentação Teórica

PUSH C

000 00 111
000 00 111
000 01 011

POP D

000 00 111
000 00 111
000 10 100

JMP #0056H

000 00 111
000 00 011
000 00 000
010 10 110

INC D, A

111 10 100

AND E, A

010 11 100

Trabalhos Correlatos

- EDSIM51
 - 8051;
 - assembly;
 - pontos de parada;

Trabalhos Correlatos

- PIC Simulator IDE
 - PIC16F, PIC12F e PIC10F;
 - basic;
 - pontos de parada;

Trabalhos Correlatos

- GPSIM
 - PIC (vários);
 - edição do circuito;
 - 25MHz em um PII 400MHz;
 - não permite a programação;
 - não permite pontos de parada;

Trabalhos Correlatos

- But How Do It Know CPU Online Model
 - UCP específica do autor;
 - 10Hz;
 - assembly;
 - visualização interna do circuito;
 - não permite a edição do circuito;
 - não permite pontos de parada;

Requisitos

- salvar e carregar projetos (RF);
- abrir arquivos já compilados (RF);
- programar e montar os programas na mesma ferramenta (RF);
- disponibilizar a depuração do software com pontos de parada (RF);

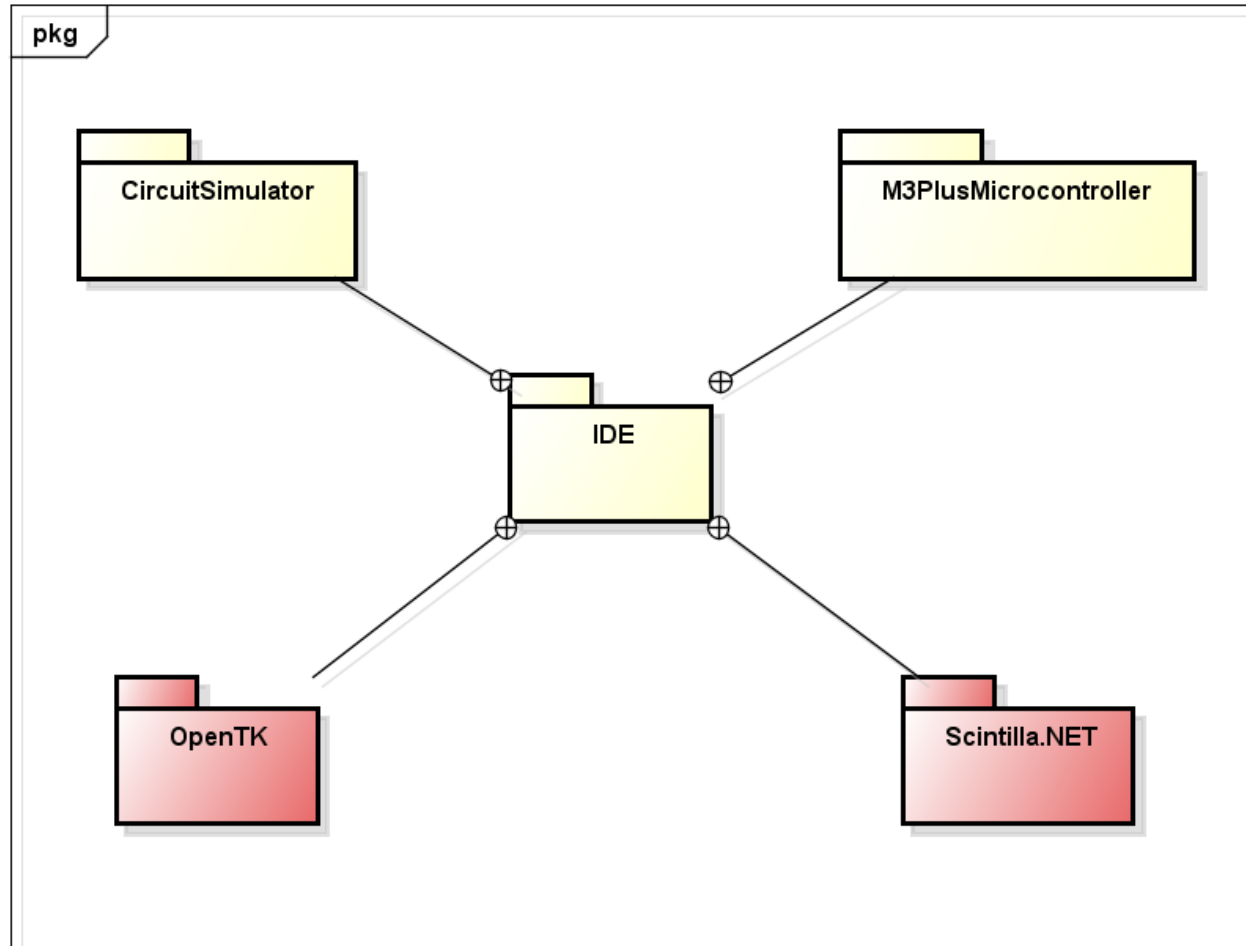
Requisitos

- possibilitar a configuração da velocidade de depuração (RF);
- permitir a construção de circuitos eletrônicos (RF);
- visualizar os sinais internos do microcontrolador (RF);

Requisitos

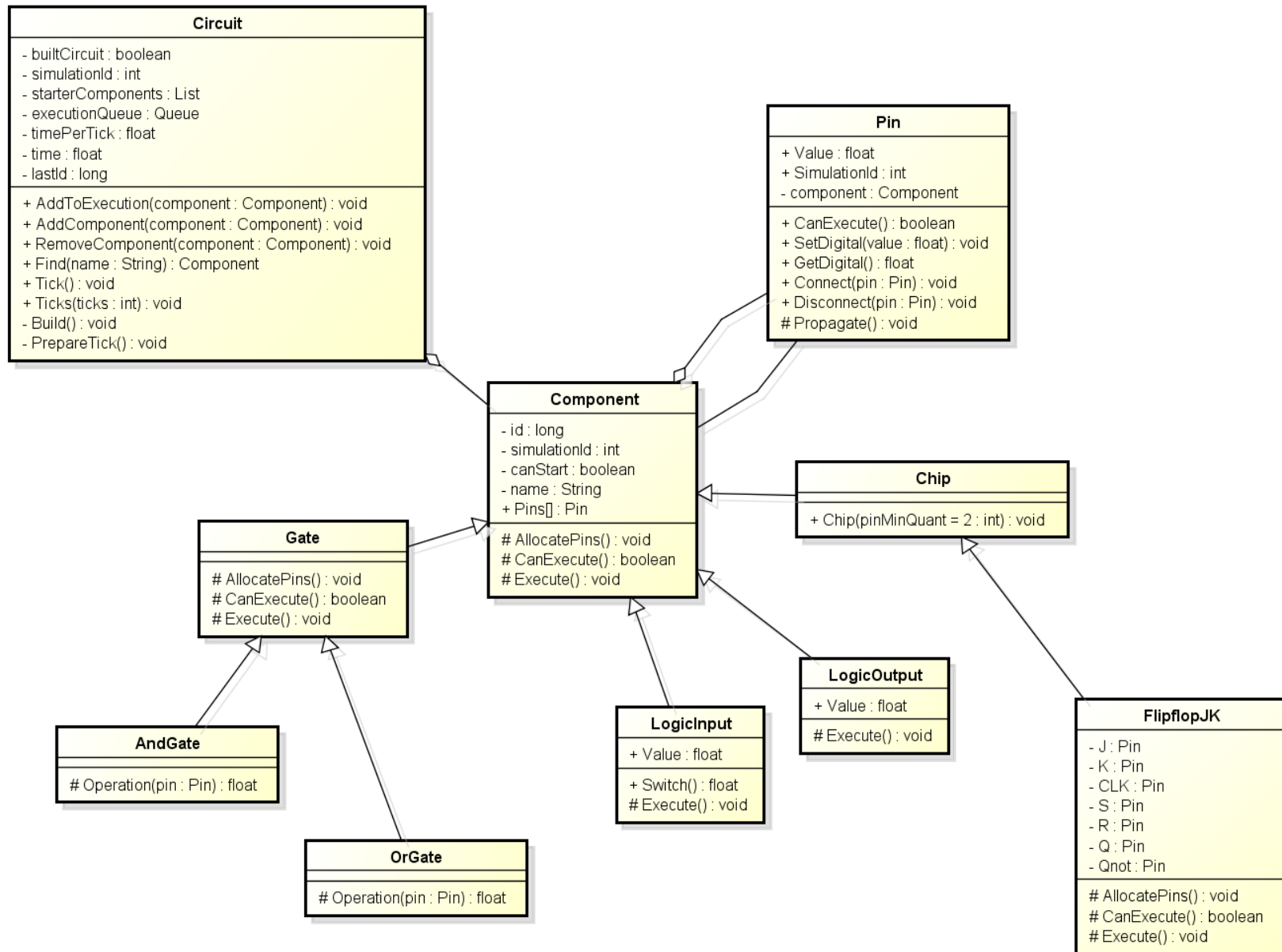
- ser implementado na linguagem C# (RNF);
- ter compatibilidade com o sistema operacional Windows 7 (RNF);
- rodar em tempo real na velocidade de clock de 1MHz (RNF);

Especificação

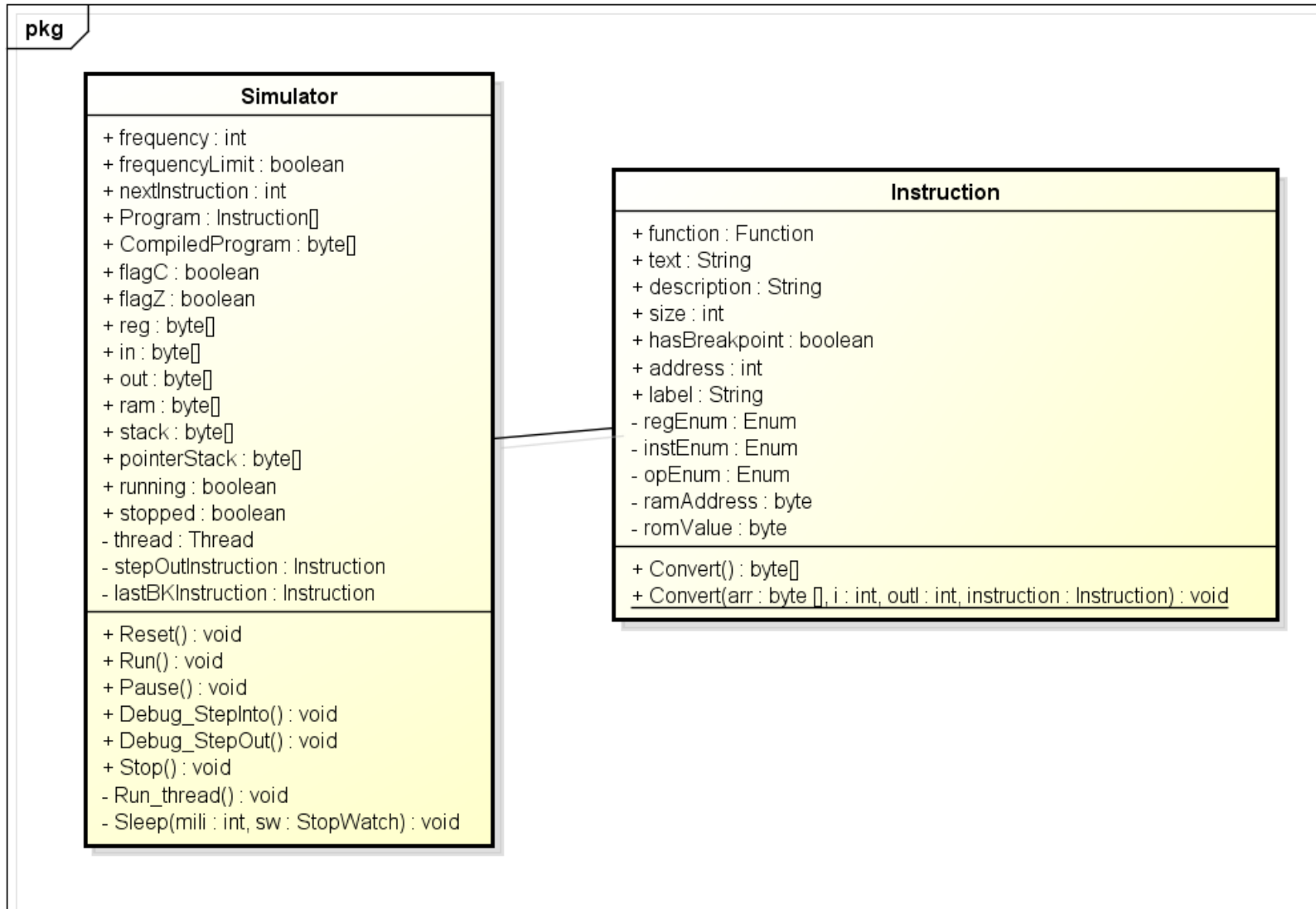


powered by Astah

Especificação - Circuit Simulator



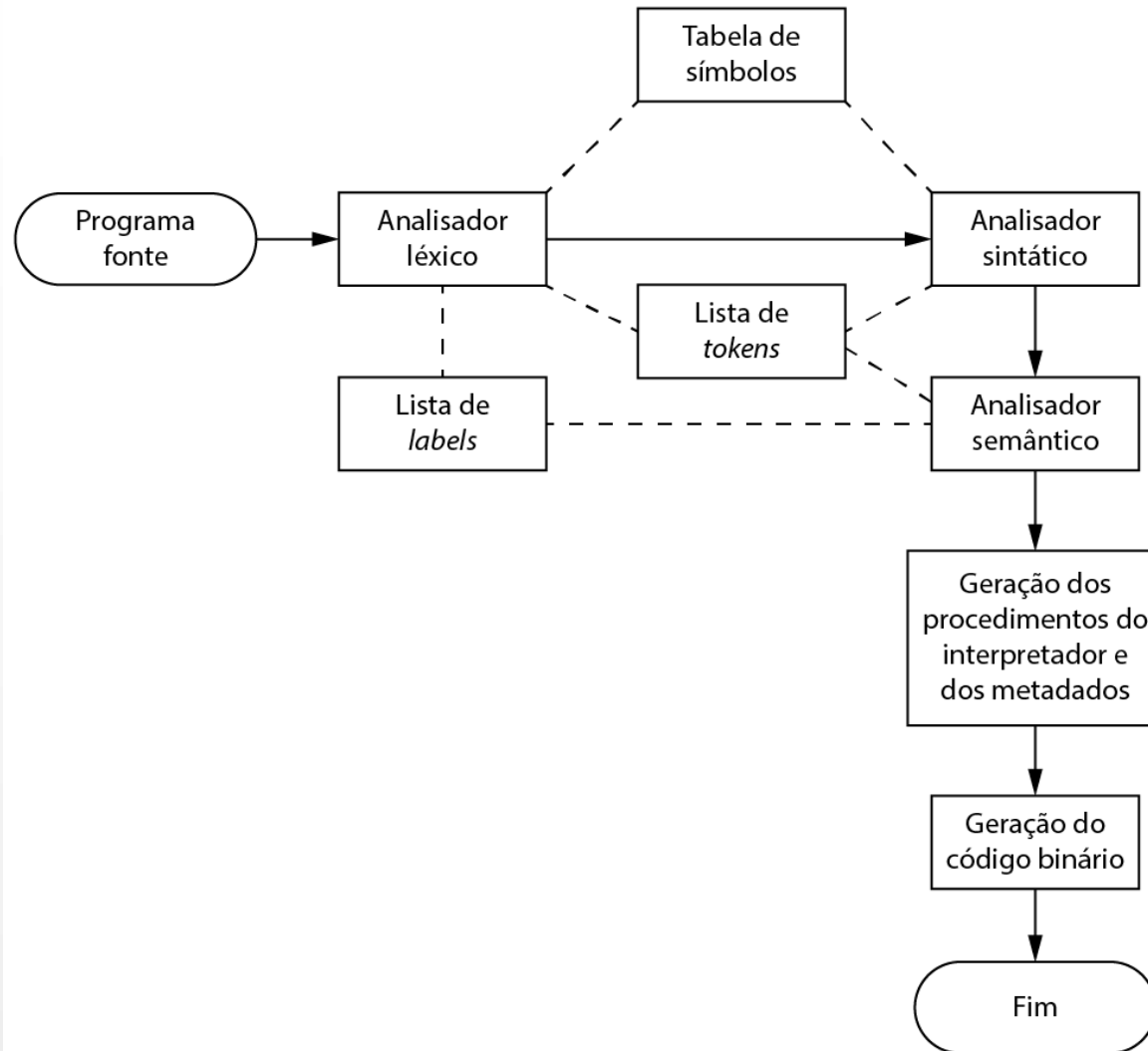
Especificação – M3PlusMicrocontroller



Implementação - Montador

- Montador
 - Análise léxica
 - Análise sintática
 - Análise semântica

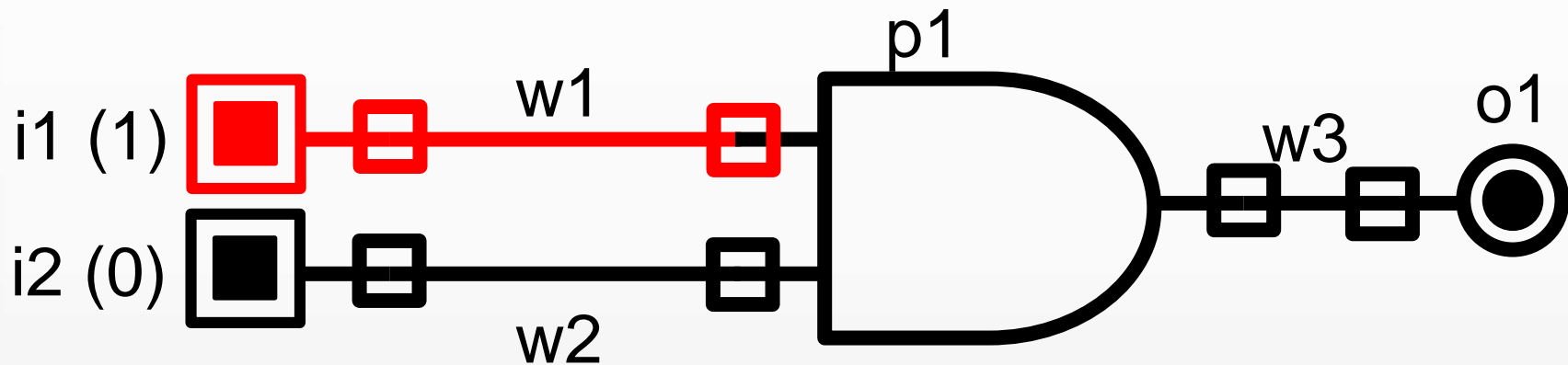
Implementação - Montador



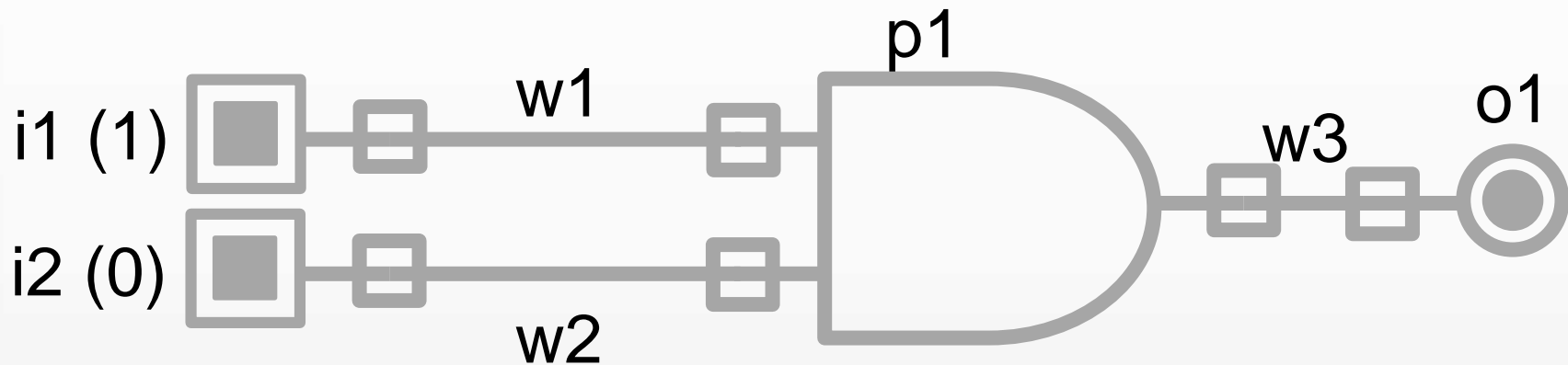
Implementação - Montador

```
1 public delegate void Function();
2 public class Instruction {
3     public Function function = delegate() { };
4     //...
5     //Instrução MOV B, A
6     instruction.function = delegate () {
7         simulator.Reg[0] = simulator.Reg[1];
8         simulator.Flag_Z = simulator.Reg[0] == 0;
9         simulator.Flag_C = false;
10    };
11    //...
12    public class Simulator{
13        Instruction[] Instructions;
14        //...
15    }
```

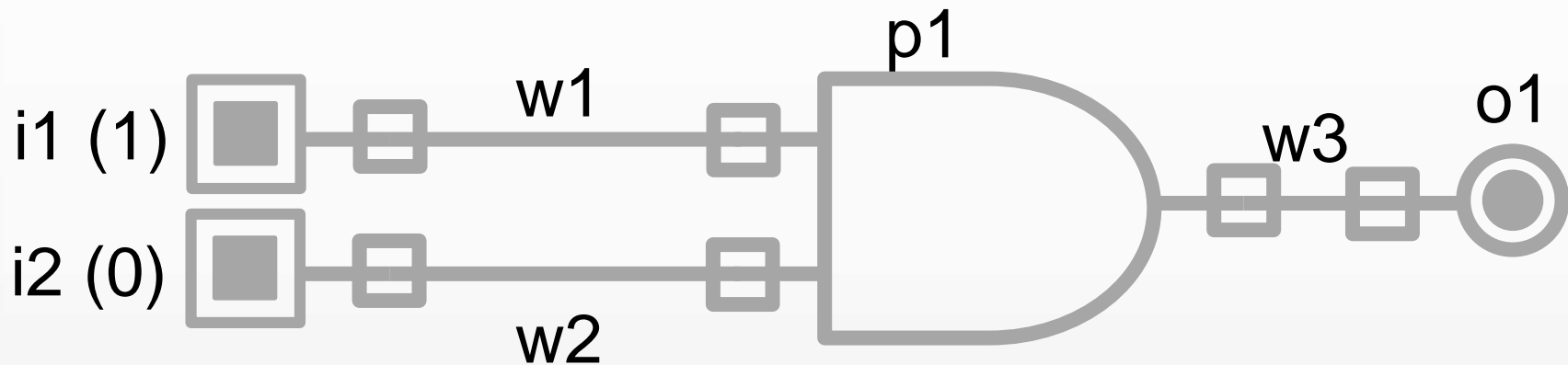
Implementação - Circuito



Implementação - Circuito



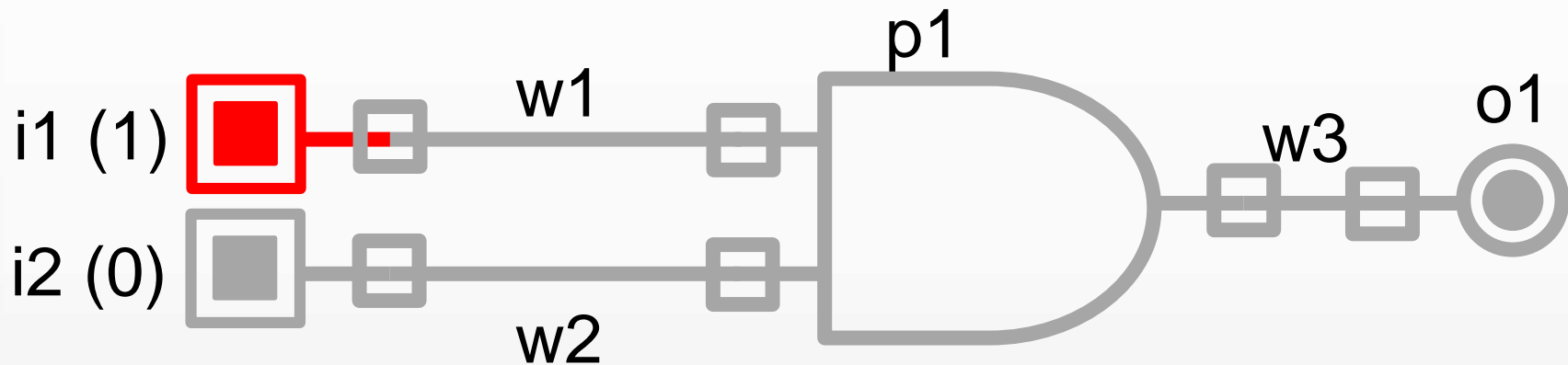
Implementação - Circuito



Fila de execução:

- $i1$
- $i2$

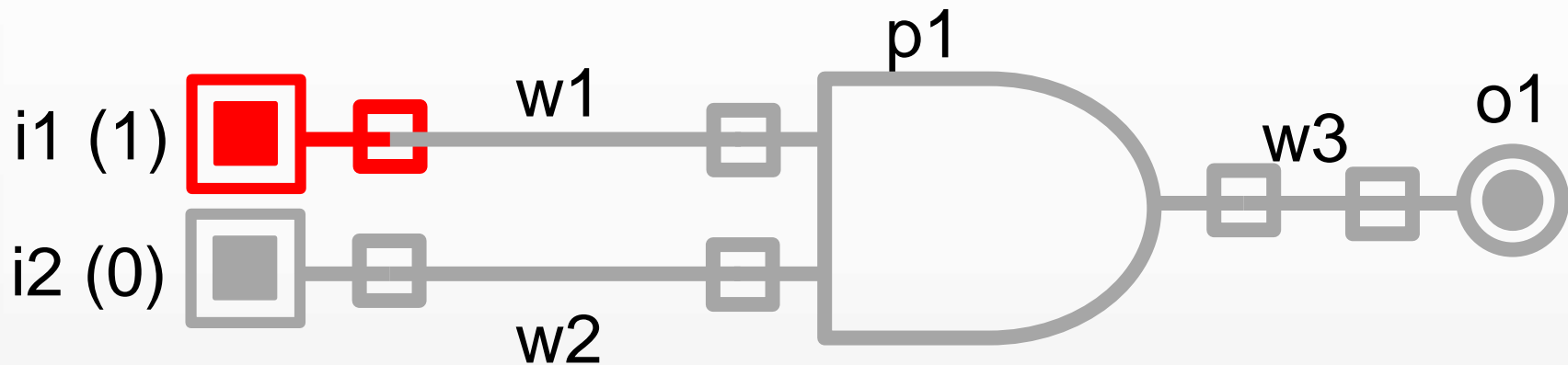
Implementação - Circuito



Fila de execução:

- i2

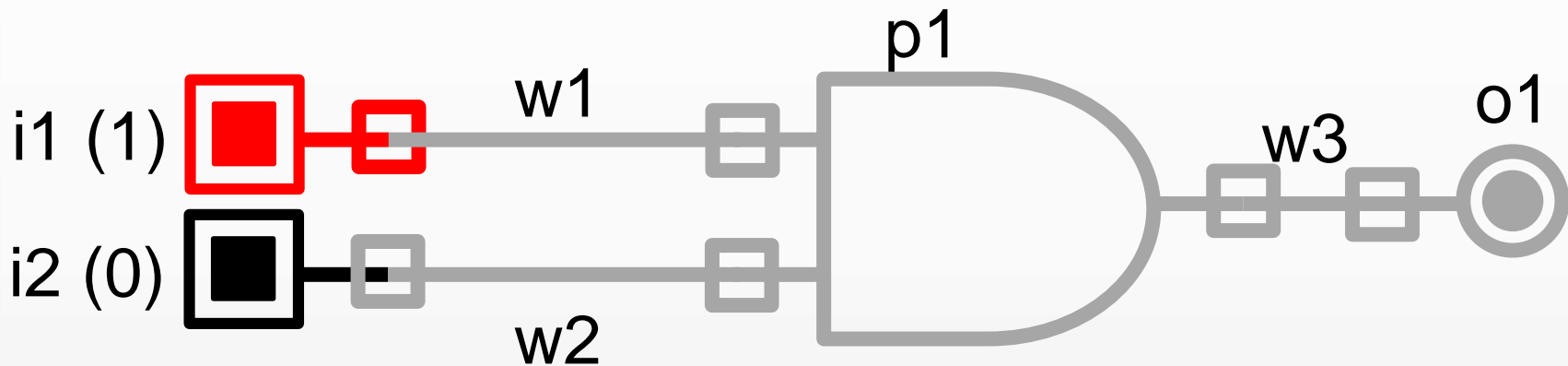
Implementação - Circuito



Fila de execução:

- i2
- w1

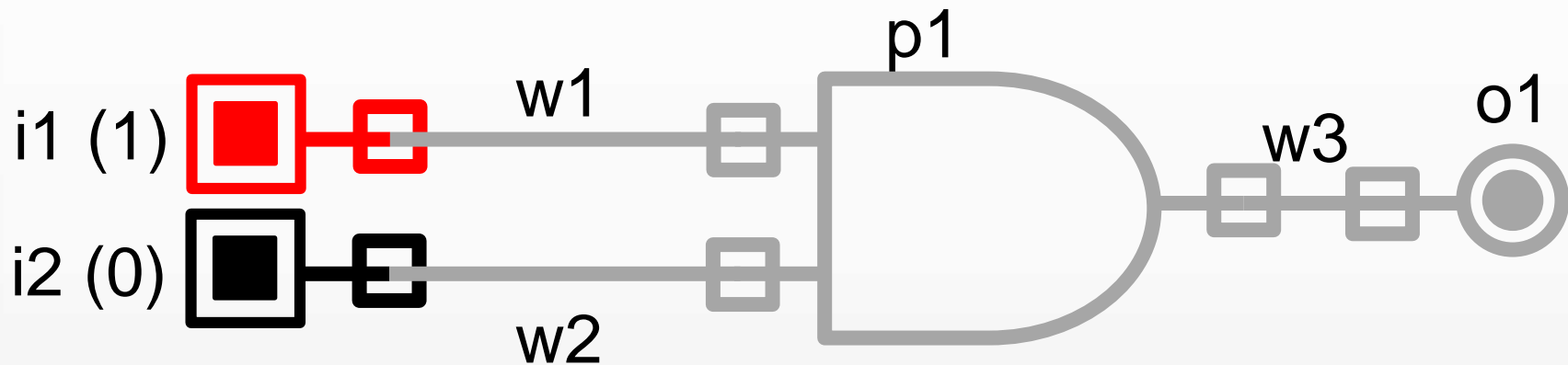
Implementação - Circuito



Fila de execução:

- w1

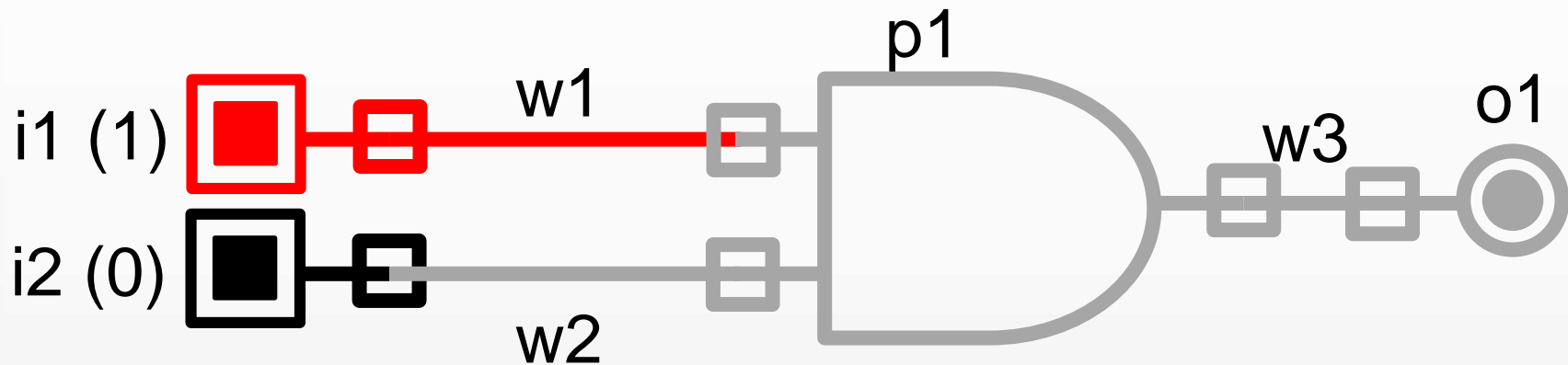
Implementação - Circuito



Fila de execução:

- w1
- w2

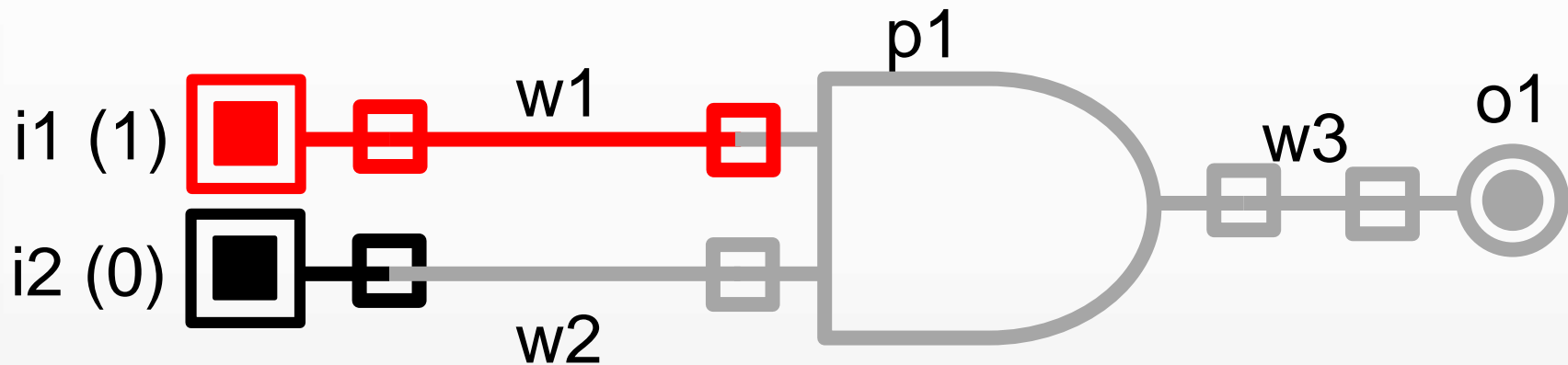
Implementação - Circuito



Fila de execução:

- w2

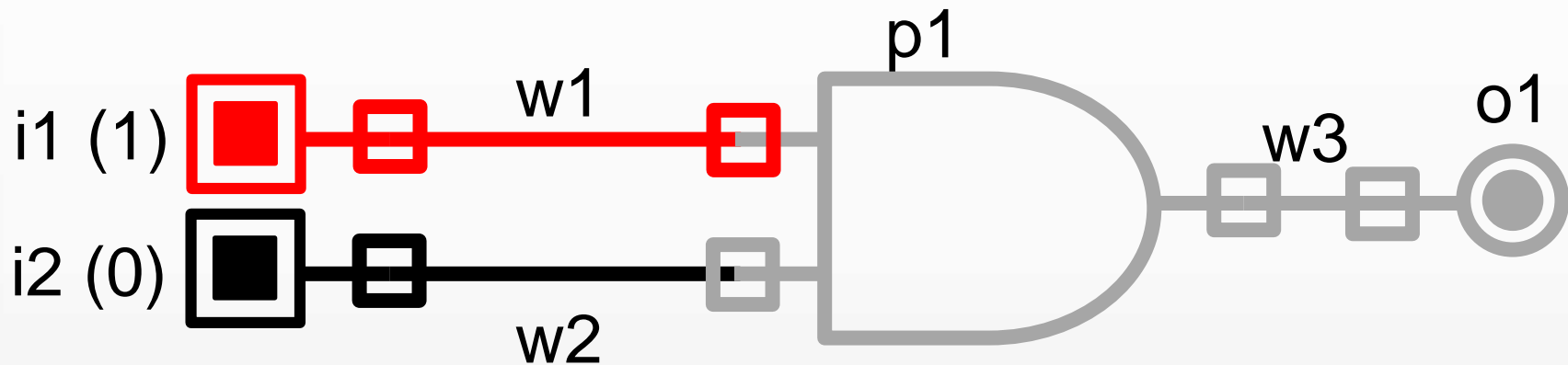
Implementação - Circuito



Fila de execução:

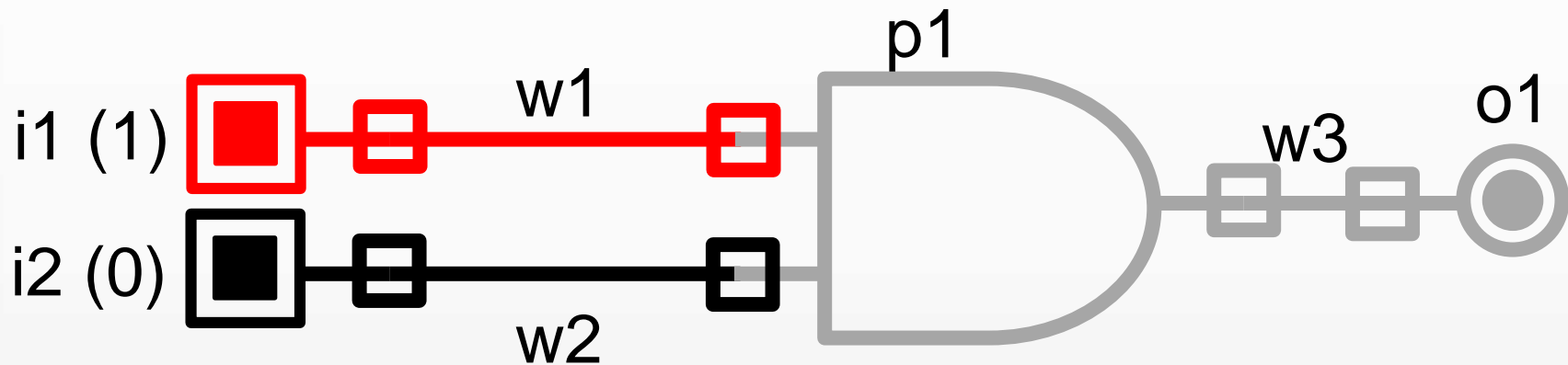
- w2

Implementação - Circuito



Fila de execução:

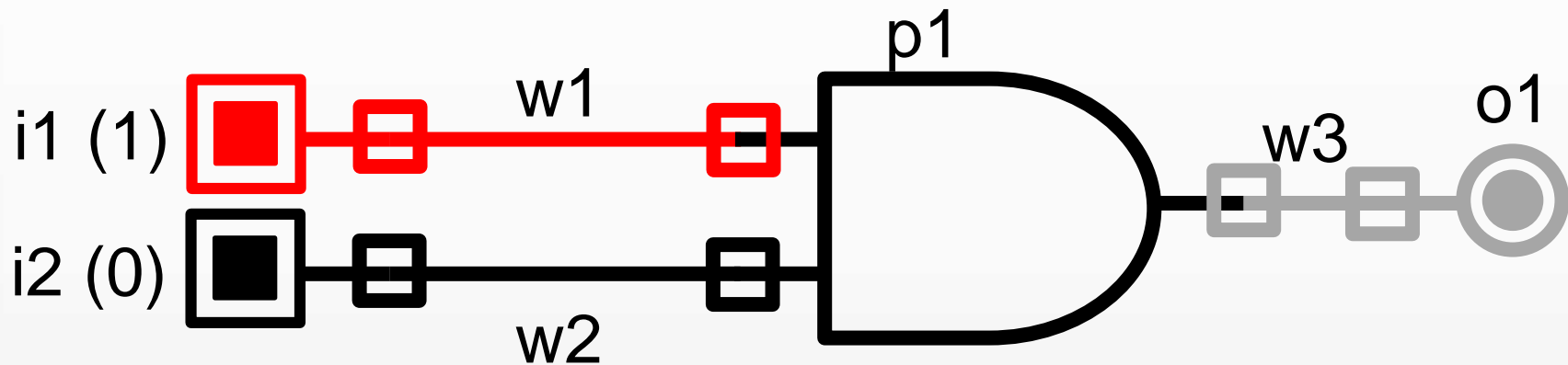
Implementação - Circuito



Fila de execução:

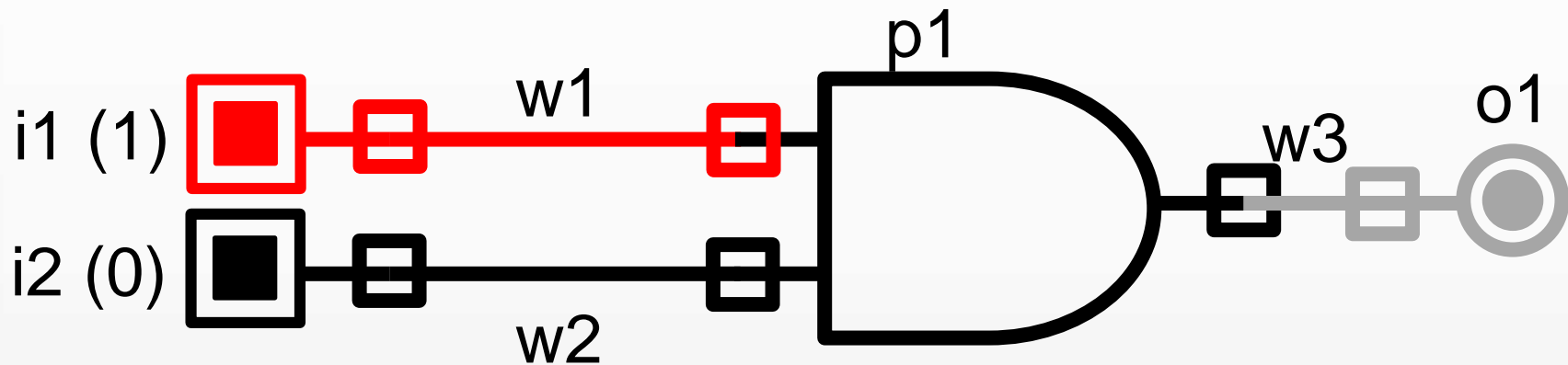
- $p1$

Implementação - Circuito



Fila de execução:

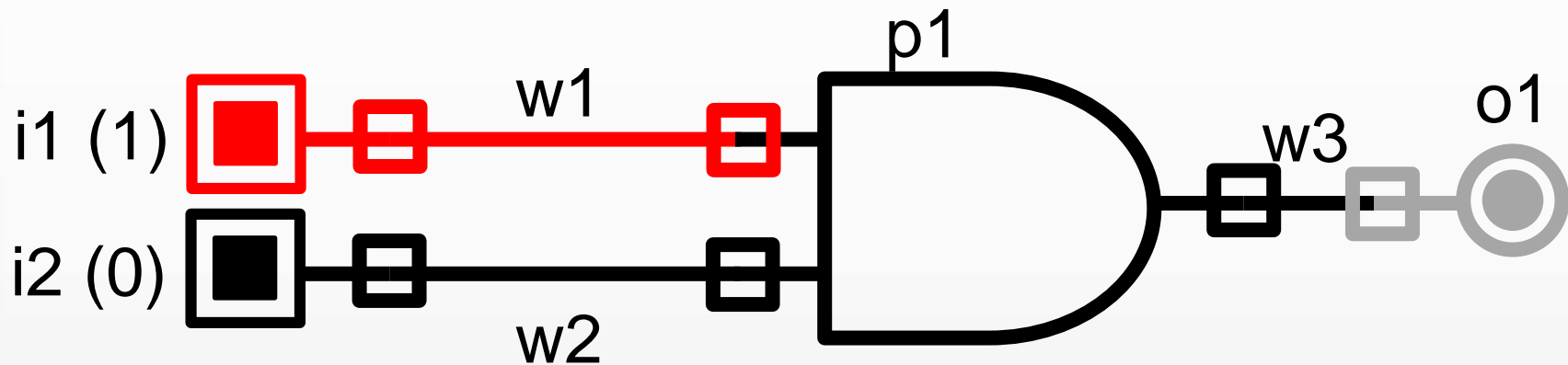
Implementação - Circuito



Fila de execução:

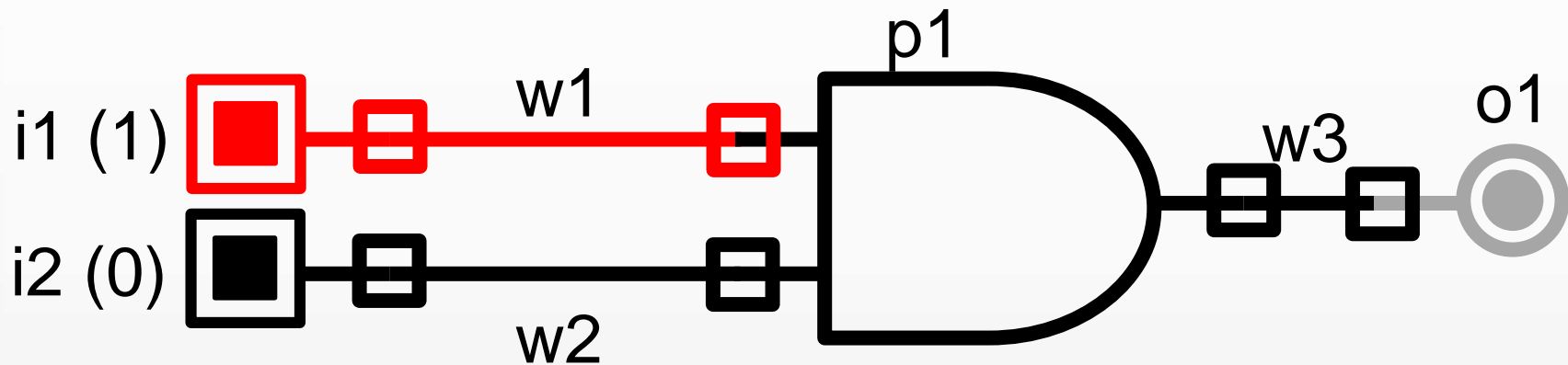
- $w3$

Implementação - Circuito



Fila de execução:

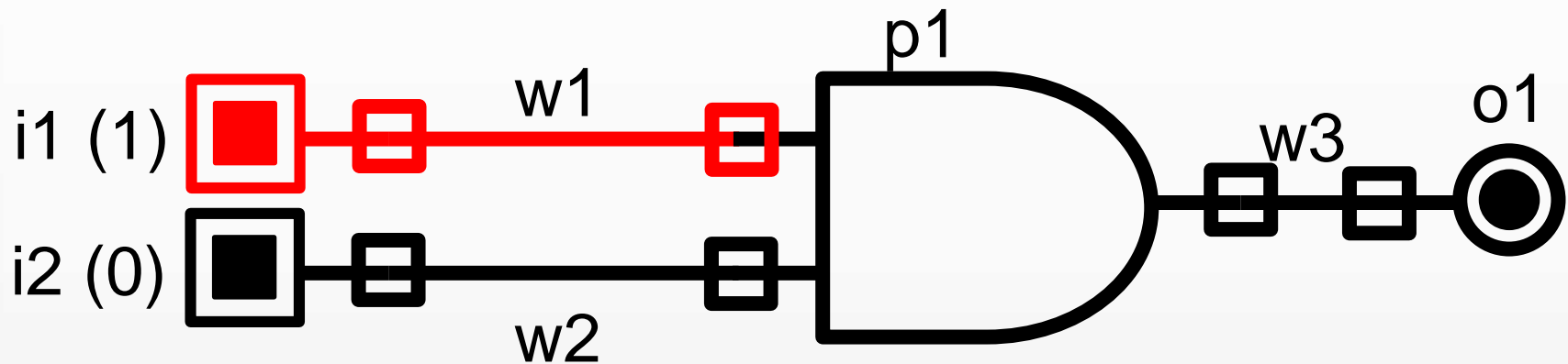
Implementação - Circuito



Fila de execução:

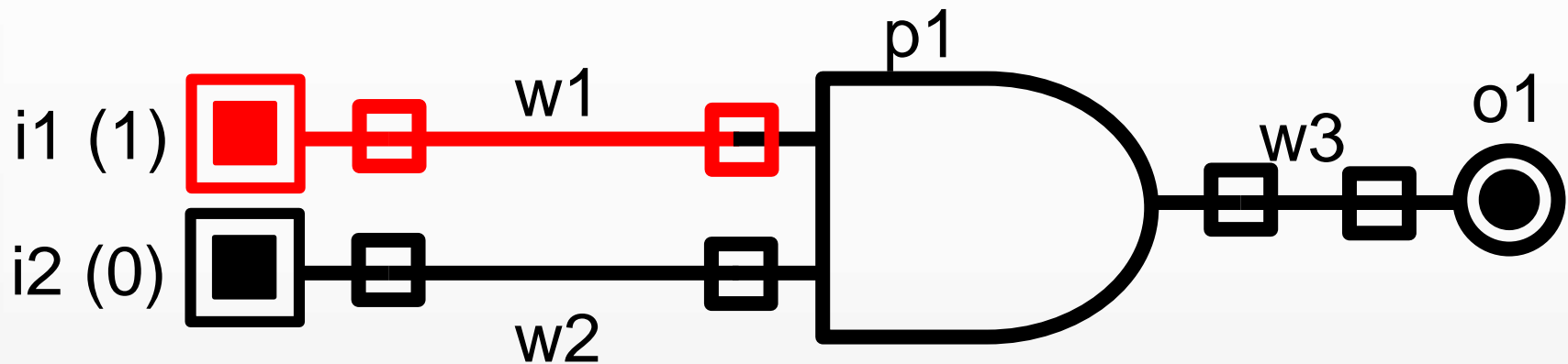
- $o1$

Implementação - Circuito



Fila de execução:

Implementação - Circuito



Fim da execução

Implementação - Flipflop

```
1 base.Execute();
2 if (S.GetDigital() == Pin.HIGH) { //S = 1
3     Q.Value = Pin.HIGH;
4     Qnot.Value = Pin.LOW;
5 } else if (R.GetDigital() == Pin.HIGH) { // R = 1
6     Q.Value = Pin.LOW;
7     Qnot.Value = Pin.HIGH;
8 } else if (CLK.Value == Pin.LOW && lastClk == Pin.HIGH) { //Clock desc
9     if (J.GetDigital() == Pin.HIGH) {
10        if (K.GetDigital() == Pin.HIGH) { // J = 1, K = 1
11            Q.Value = Q.Neg();
12            Qnot.Value = Qnot.Neg();
13        } else { // J = 1, K = 0
14            Q.Value = Pin.HIGH;
15            Qnot.Value = Pin.LOW;
16        }
17    } else {
18        if (K.GetDigital() == Pin.HIGH) { // J = 0, K = 1
19            Q.Value = Pin.LOW;
20            Qnot.Value = Pin.HIGH;
21        } else { // J = 0, K = 0
22            }
23    }
24 }
25 lastClk = CLK.Value;
26 Q.Propagate();
27 Qnot.Propagate();
```


Operacionalidade da Implementação

PC: ▾

SP: ▾

Frequência

Limitar frequência

Simulação interna

▾

IPS real: 0 IPS

Registradores

A ▾

B ▾

C ▾

D ▾

E ▾

Flags

Carry

Zero

Entradas

Ativar edição das entradas

IN0 ▾

IN1 ▾

IN2 ▾

IN3 ▾

Saídas (somente leitura)

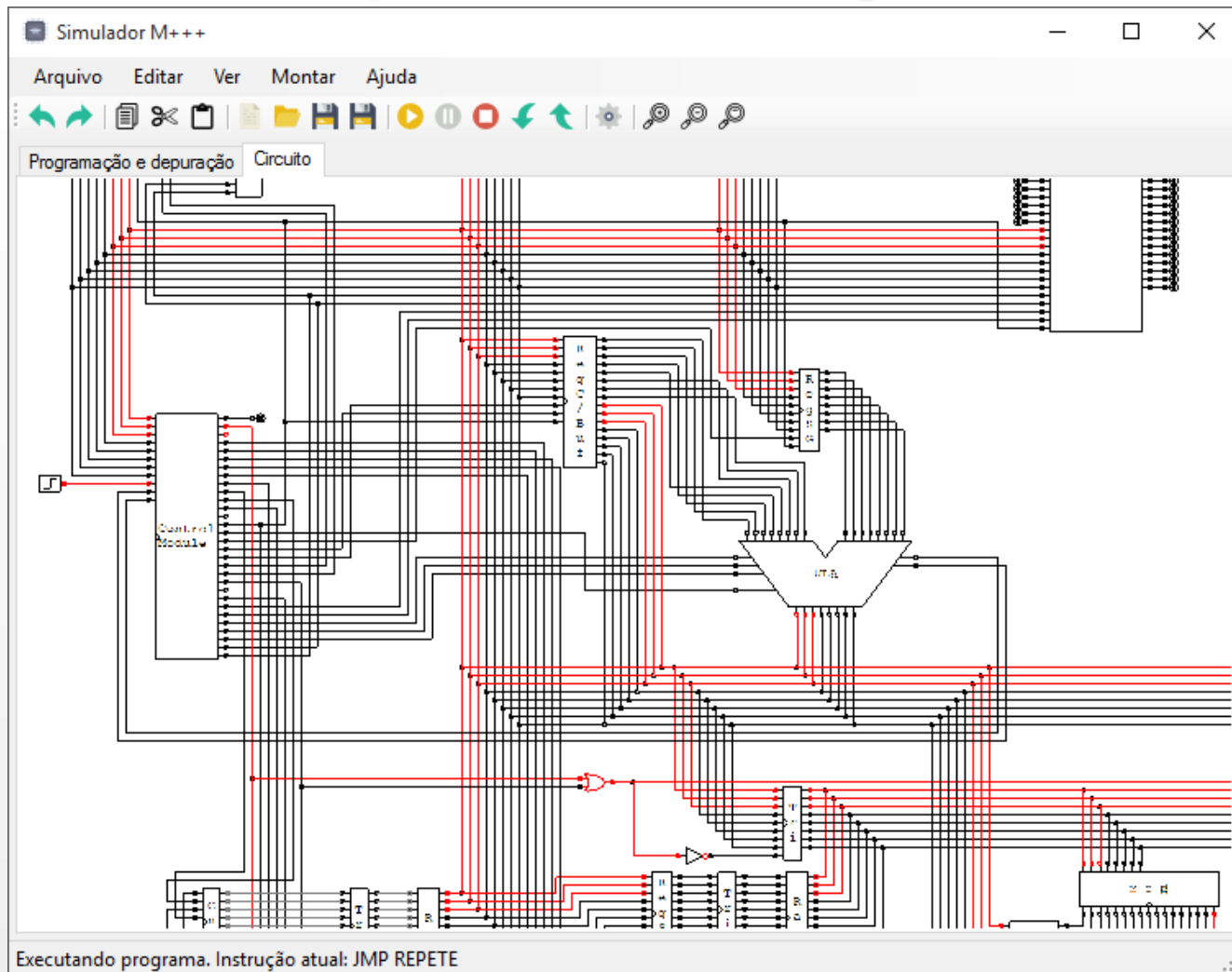
OUT0 ▾

OUT1 ▾

OUT2 ▾

OUT3 ▾

Operacionalidade da Implementação



Resultados e Discussões

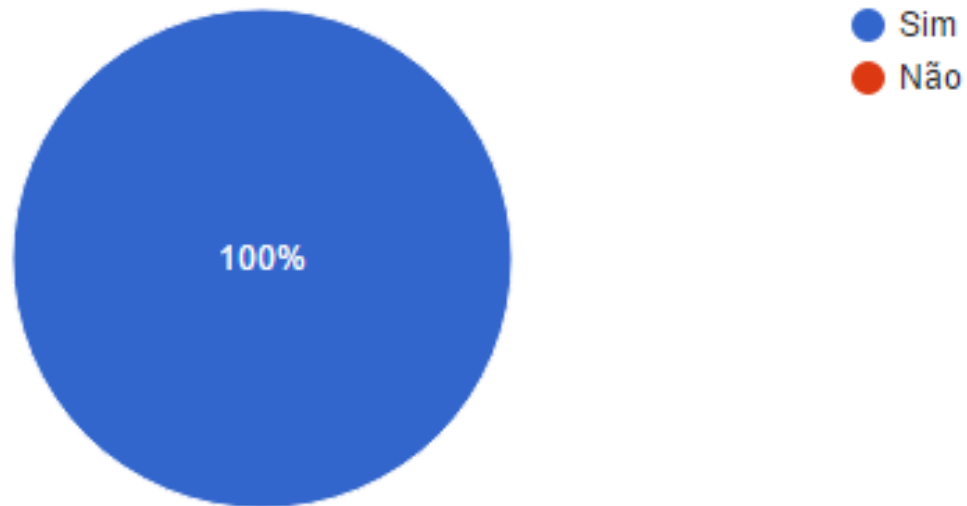
	EDSIM51	PIC Simulator IDE	GPSIM	But How Do It Know	Este projeto
Velocidade da simulação	Não deixa claro	Não deixa claro	25MHz @ PII 400MHz	10Hz	120MIPS @ i7 4790k 4.0GHz
Permite a depuração e inserção de <i>breakpoints</i>	Sim	Sim	Sim	Não	Sim
Permite programar o microcontrolador diretamente na ferramenta	Sim	Sim	Não	Sim	Sim
Possibilita a montagem de circuitos eletrônicos	Não	Não	Sim	Não	Sim
Quantidade de componentes eletrônicos	9	24	11	0	14*

* Desconsiderando os componentes internos do microcontrolador.

Resultados e Discussões

O simulador contribuiu para os seus estudos?

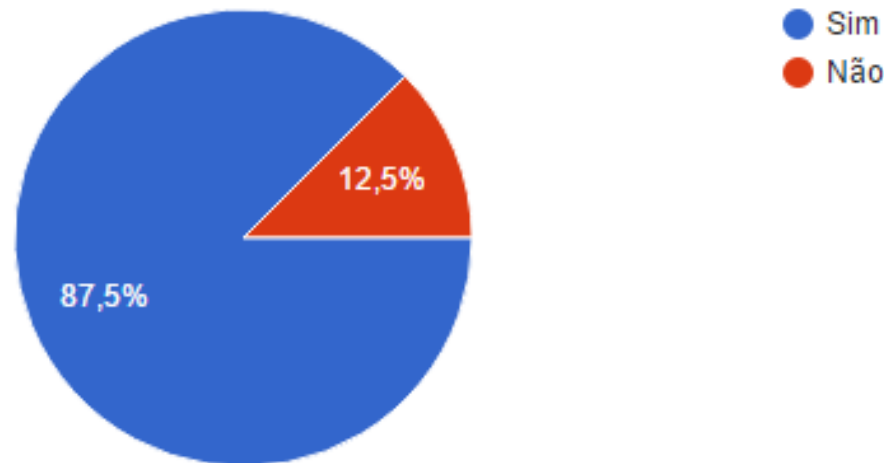
24 respostas



Resultados e Discussões

Você teve vontade de aprender mais sobre a arquitetura de computadores?

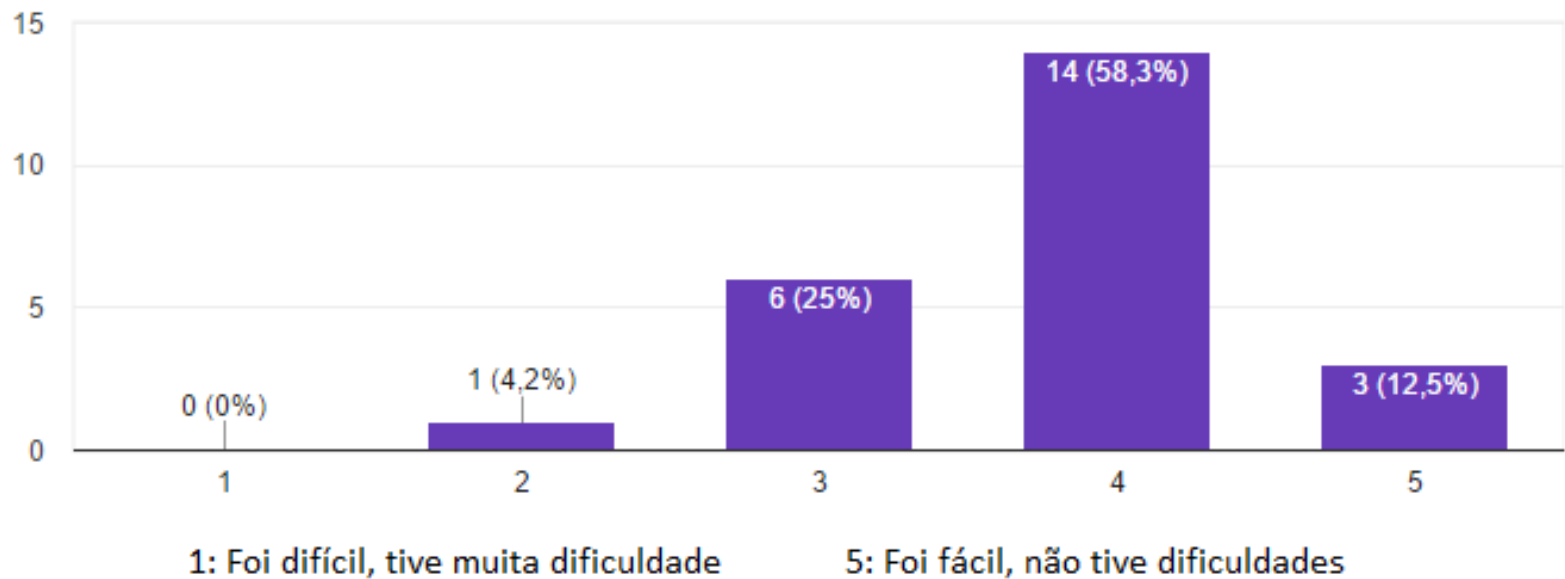
24 respostas



Resultados e Discussões

Em uma escala de 1 a 5, o quão fácil foi operar no simulador?

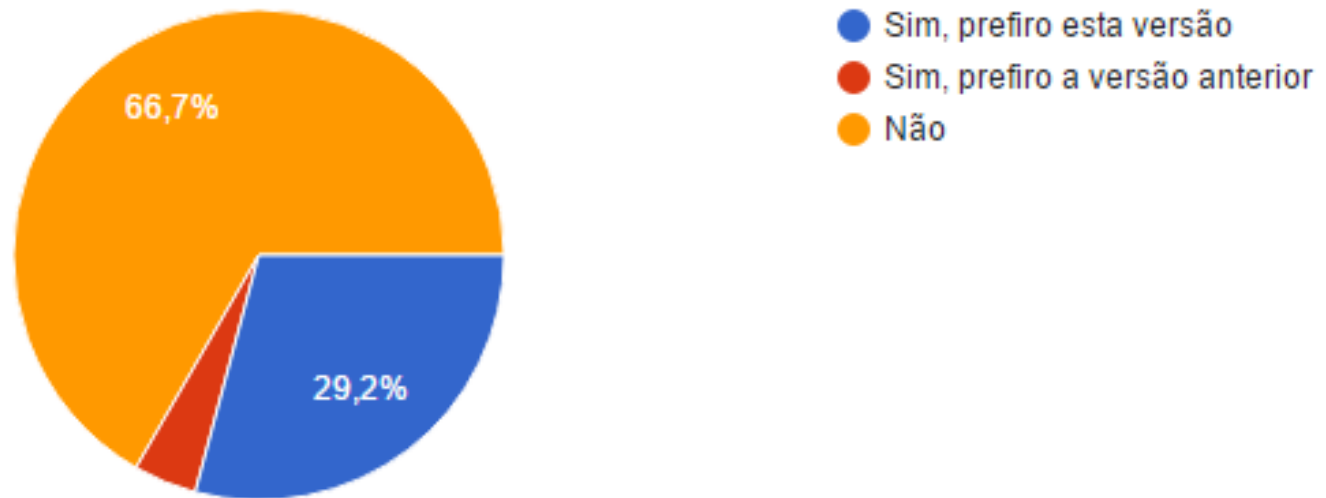
24 respostas



Resultados e Discussões

Você chegou a utilizar a versão anterior desenvolvida no Logisim?

24 respostas



Conclusões e Sugestões

- grande vantagem na velocidade da simulação;
- pontos de parada;
- uma única interface;

Conclusões e Sugestões

- limitação na edição dos componentes;

Conclusões e Sugestões

- aumentar a quantidade de componentes eletrônicos;
- permitir a criação de sub-circuitos ou macros, parecido com software Logisim;
- ampliar a quantidade de instruções disponíveis no microcontrolador;

Conclusões e Sugestões

- corrigir alguns problemas não solucionados que fecham o software e não puderam ser reproduzidos neste trabalho;
- tornar o log mais amigável visualmente;
- tornar o circuito mais amigável visualmente.