



# Apresentação Final

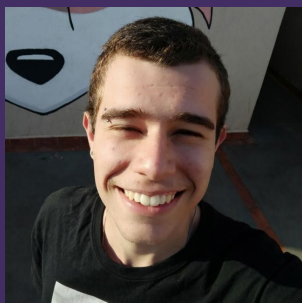
Organização e Arquitetura  
de  
Computadores

Grupo 8

# Organização e Arquitetura de Computadores

Grupo 8

# INTEGRANTES



**Nicolas C. Caldeira**

Engenharia de Computação

Ingressante na Engenharia de Computação em 2021, tem interesse em trabalhar com Desenvolvimento Web



**Matheus dos S. Inês**

Engenharia de Computação

Transferido da Engenharia de Materiais (021), tem interesse em trabalhar com Produção Musical e Cinematográfica



**Mateus Goes**

Engenharia de Computação

Transferido da Engenharia de Materiais (021), tem interesse em trabalhar com Mercado Financeiro



**Bárbara F. Madera**

Engenharia de Computação

Ingressante na Engenharia de Computação em 2020, tem interesse em trabalhar com a aplicação da computação em robótica de reabilitação

# PARTE 1

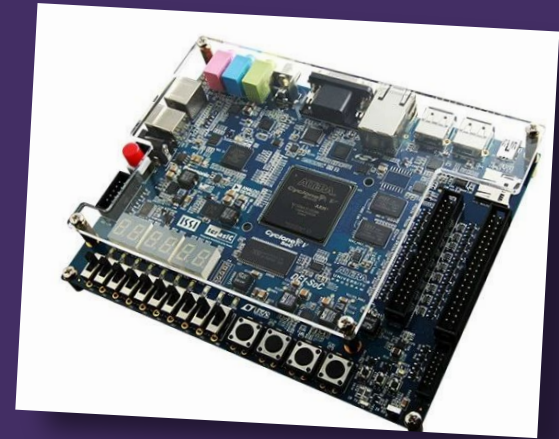
## Processador em SystemVerilog



# Ideia



- Primeiro contato com Verilog em Sistemas Digitais (SEL0628)
- Adquirir novos conhecimentos e Agregar e aplicar conceitos da graduação
- Máquina de estado: Possibilidade de visualizar diretamente (FPGA)



# Ideia

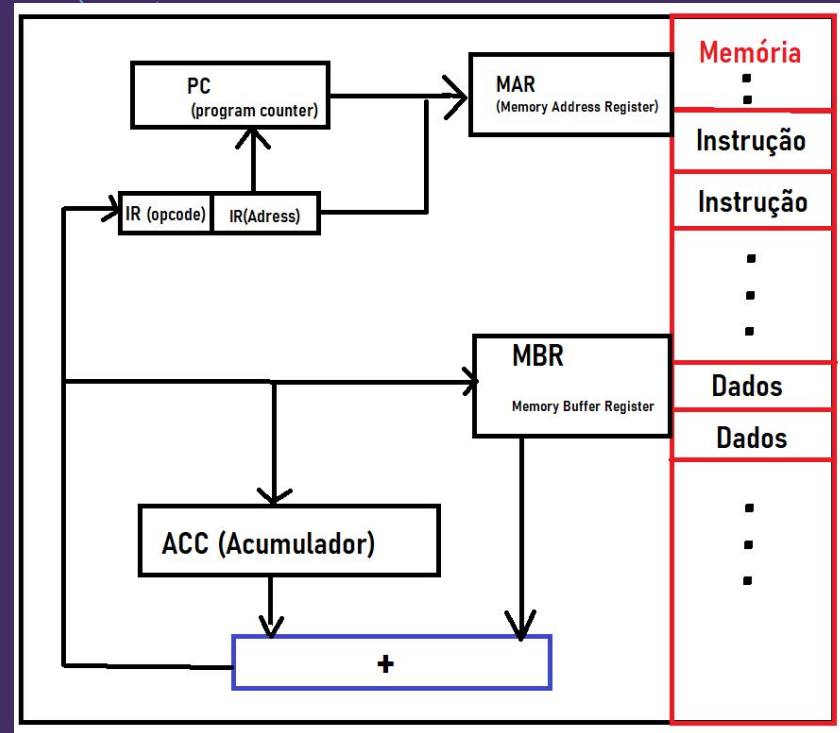


- Bagagem de conhecimento sobre outras linguagens
- Quartus Prime
- Visualização em pequena escala de algo muito complexo



# Arquitetura do Processador

- 8 bits
- 4 instruções
- 2 formatos
- Arquitetura Von Neumann



# Organização de memória

## Organização de memória em 128 endereços

- 16 variáveis possíveis
- A parte vazia pode ser usada para armazenar vídeo ou acoplar gráfico no processador no kit FPGA

### Instruções

Endereço	Palavras
0 0 0 0 0 0 0 0	$2^7=128$
0 1 1 1 1 1 1 1	

### Dados

Endereço	Palavras
1 1 1 1 0 0 0 0	$2^4=16$
1 1 1 1 1 1 1 1	

### Vazia

Endereço	Palavras
1 1 1 1 0 0 0 0	$2^7 - 2^4=112$
1 1 1 1 1 1 1 1	

# Conjunto de instruções - 8 bits - 4 instruções - 2 formatos

## Tipo 1

Endereço: 0x1111\_endereçoX

7	6	5	4	3	2	1	0	bits
0	0	0	0	endereço X				não usada
0	0	0	1	endereço X				não usada
0	0	1	0	endereço X				não usada
0	0	1	1	endereço X				STORE
0	1	0	0	endereço X				LOAD
0	1	0	1	endereço X				ADD
0	1	1	0	endereço X				não usada
0	1	1	1	endereço X				não usada

## Tipo 2

Endereço: 0x0\_endereçoY

7	6	5	4	3	2	1	0	bits
1				endereço Y (+7 bits)			Jump	



# O CÓDIGO

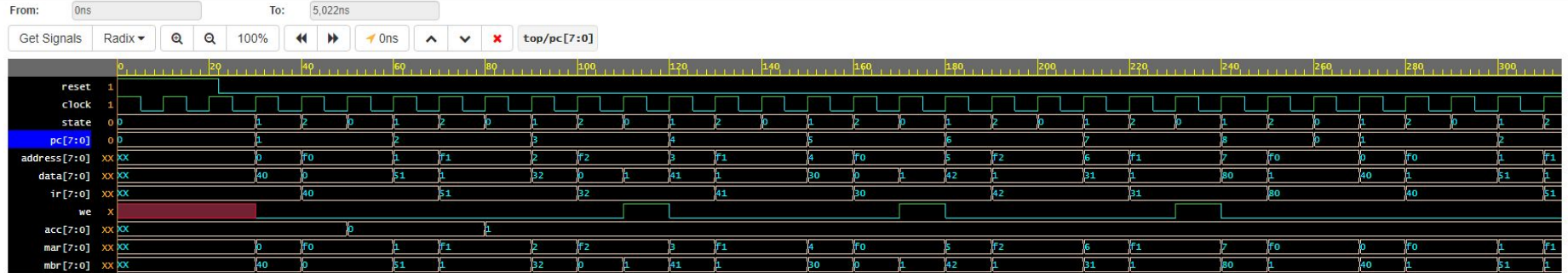
Explicação

# Código

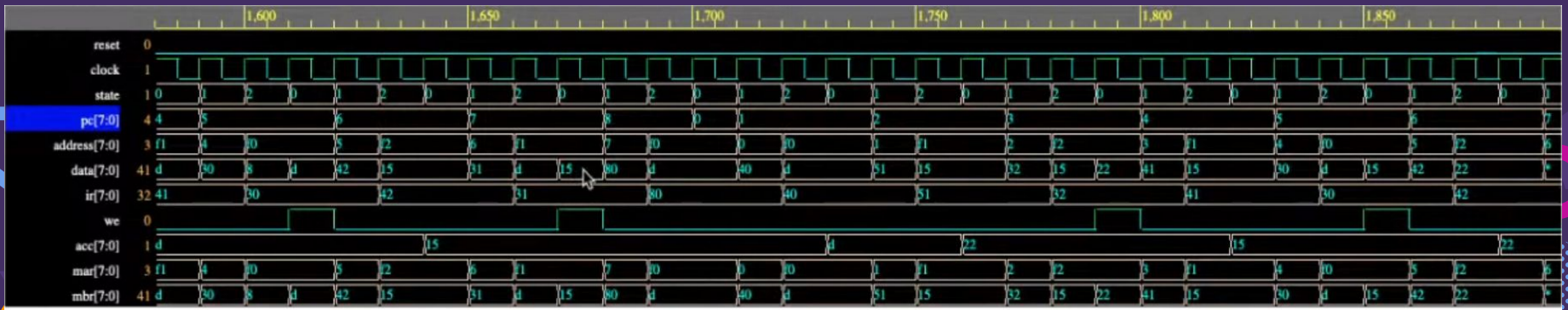
```
1 module cpu(  
2   input clock, reset,  
3   inout [7:0] mbr,  
4   output logic we,  
5   output logic [7:0] mar, pc, ir);  
6  
7   typedef enum logic [1:0] {FETCH, DECODE, EXECUTE} statetype;  
8   statetype state, nextstate;  
9  
10  logic [7:0] acc;  
11  
12  always @(posedge clock or posedge reset)  
13  begin  
14    if (reset) begin  
15      pc = 'b0;  
16      state <= FETCH;  
17    end  
18    else begin  
19      case(state)  
20        FETCH: begin  
21          we <= 0; //ciclo de busca (não escreve na memória)  
22          pc <= pc + 1; //incrementa o pc (program counter)  
23          mar <= pc; //coloca o endereço como sendo o PC pra acessar a memória - feitas ao mesmo tempo  
24        end  
25        DECODE: begin  
26          ir = mbr; //traz o dado da memória  
27          mar <= {4'b1111, ir[3:0]}; //ao acessar a memória eu pego os 4 bits finais e preencho com 1111  
28        end  
29        EXECUTE: begin  
30          if (ir[7] == 1'b1) // JUMP: Se o bit que está mais a esquerda for 1  
31            pc <= {1'b0, ir[6:0]}; //pega o PC e salta pros 7 bits da instrução concatenado com 1 zero  
32          else if (ir[7:4] == 4'b0100) // LOAD : pega o dado da memória e coloca no acumulador  
33            acc <= mbr;  
34          else if (ir[7:4] == 4'b0101) // ADD: recebe ele mesmo + valor da memória  
35            acc <= acc + mbr;  
36          else if (ir[7:4] == 4'b0111) // SUB: recebe ele mesmo - valor da memória  
37            acc <= acc - mbr;  
38          else if (ir[7:4] == 4'b0011) // STORE: Habilita o 1 no write enable  
39            we <= 1'b1; //habilita a escrita na memória  
40        end  
41      endcase  
42      state <= nextstate;  
43    end  
44  end
```

# Resultados

EPWave



Note: To revert to EPWave opening in a new browser window, set that option on your user page.



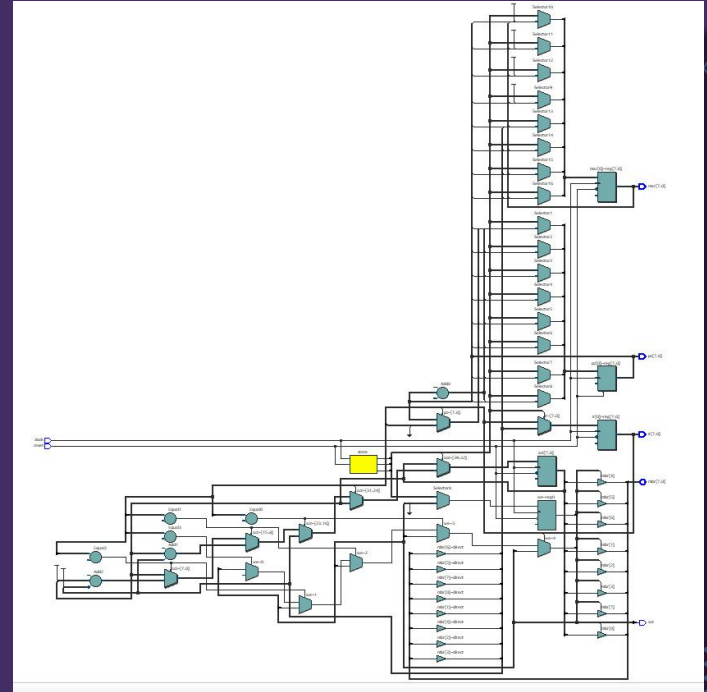
# Flow Summary (Quartus)

- Fornece informações técnicas muitas vezes relevantes
- Indica se o projeto excedeu os recursos disponibilizados
- Pinos, bits de memória, blocos de processamento de sinal digital (DSP) e loops de bloqueio de fase (PLLs)

Flow Summary	
<<Filter>>	
Flow Status	Successful - Tue Dec 20 22:16:37 2022
Quartus Prime Version	21.1.1 Build 850 06/23/2022 SJ Lite Edition
Revision Name	Org_Arq
Top-level Entity Name	cpu
Family	Cyclone IV GX
Total logic elements	61 / 14,400 ( < 1 % )
Total registers	36
Total pins	35 / 81 ( 43 % )
Total virtual pins	0
Total memory bits	0 / 552,960 ( 0 % )
Embedded Multiplier 9-bit elements	0
Total GXB Receiver Channel PCS	0 / 2 ( 0 % )
Total GXB Receiver Channel PMA	0 / 2 ( 0 % )
Total GXB Transmitter Channel PCS	0 / 2 ( 0 % )
Total GXB Transmitter Channel PMA	0 / 2 ( 0 % )
Total PLLs	0 / 3 ( 0 % )
Device	EP4CGX15BF14C6
Timing Models	Final

# Circuito RTL (Quartus)

- Forma de visualizar um circuito digital
- Ajuda a entender o que foi escrito na linguagem escolhida
- Quartus: RTL Viewer



# PARTE 2

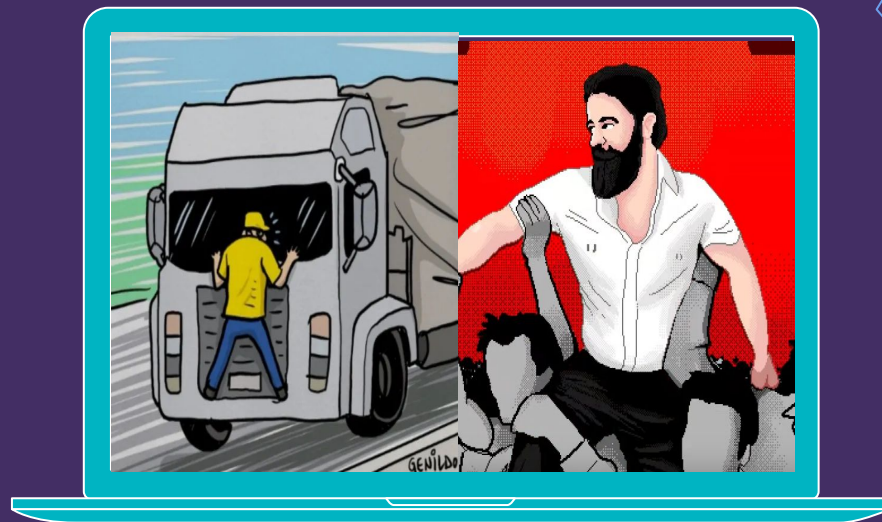
Jogo da Velha: Edição  
Patriotas x Trabalhadores



# Sobre o tema



- Jogo da Infância
- Eleições
- Renovar algo clássico
- Bom humor
- Tudo é político



# O JOGO

Demonstração

Link para o vídeo explicativo:

<https://www.youtube.com/watch?v=KHNYgREmD5g>

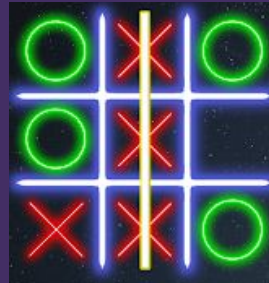


# Dificuldades



- Contato com algo novo (embora a linguagem seja antiga)
- Programação + Teoria

**+ASSEMBLY**



# Melhorias



- Tela de vitória (Patriotas)
- Maior dinamicidade para as telas (movimento)
- Melhoria do design
- Multijogos e single player



# Melhorias



- Tela de vitória (Trabalhadores)
- Maior dinamicidade para as telas (movimento)
- Melhoria do design
- Multijogos e single player



The slide features a dark purple background with various geometric shapes in the corners. Top-left: a pink circle, a dashed yellow circle, a solid purple pentagon, a blue dotted circle, and a pink outline pentagon. Top-right: a blue triangle, a dashed yellow circle, a solid purple circle, a pink and white striped circle, and a blue outline pentagon. Bottom-left: a blue outline pentagon, a pink triangle, a yellow and black striped triangle, and a dashed yellow circle. Bottom-right: a pink circle, a dashed yellow triangle, and a blue dotted circle.

Agradecemos  
pela atenção :)

BOAS FESTAS!