Design e implementação de um processador em FPGA

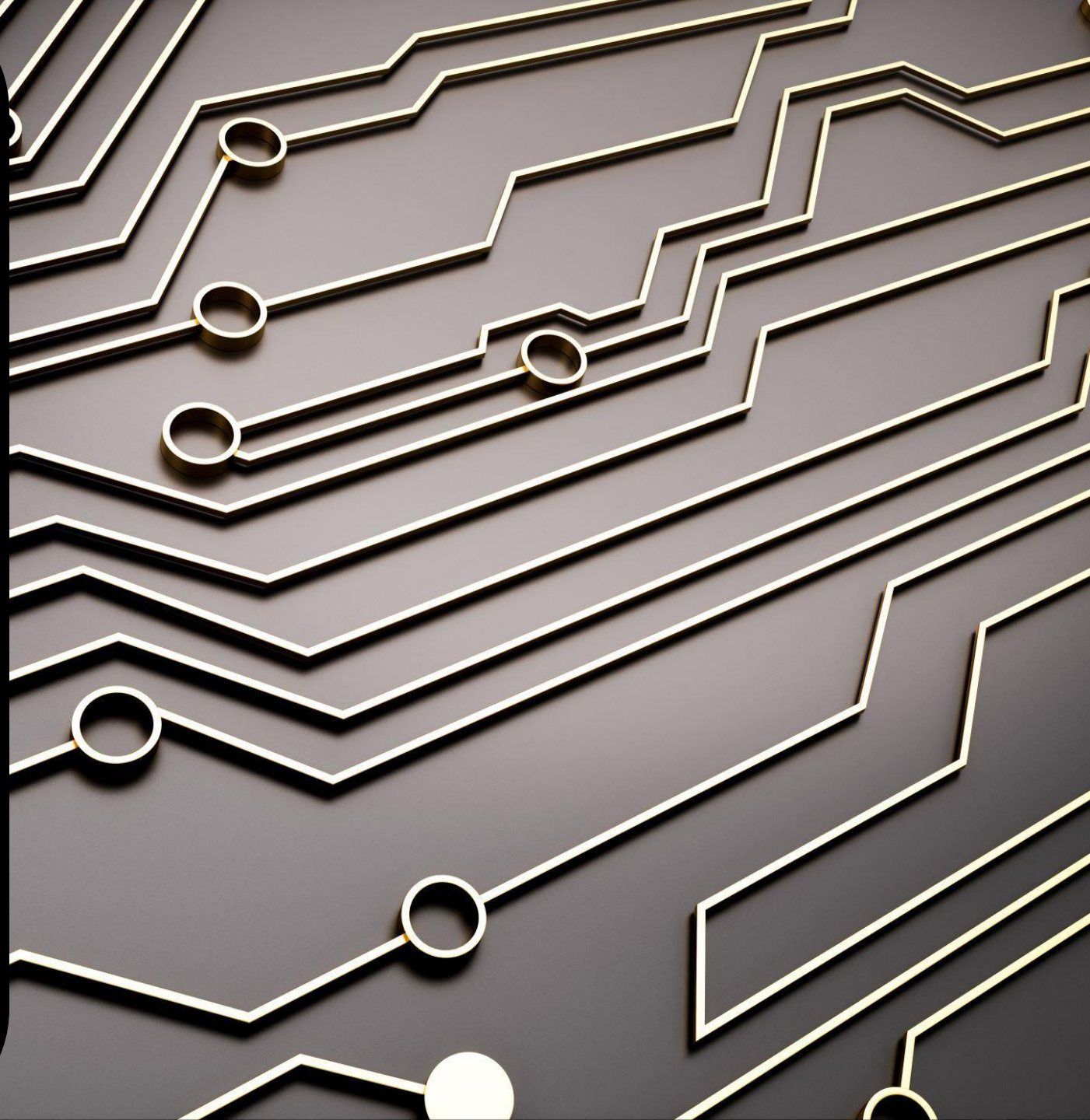
Álvaro Moisés Frate Serantola - 202402348957

Gabriel Baptista Moreira dos Santos - 202403852497

Gabriel Danilo Rosalino Batista - 202403474085

João Antonio Curtis Lopes - 202403033577

Rogério Samuel Valentim da Silva - 202402348973



FPGA

- FPGA (Field-Programmable Gate Array);
  - dispositivo semicondutor que pode ser programado (ou configurado) ;
  - conjunto de blocos lógicos interconectados por meio de uma matriz programável;
  - Possibilidade de criar um processador personalizado.

- Flexibilidade:
  - Possibilidade de reprogramar conforme necessário.
- Desempenho personalizado:
  - Processadores FPGA podem ser personalizados para desempenho em tarefas específicas.
- Baixo custo:
  - Mais econômicos quando comparados a desenvolver um processador personalizado em silício.
- Paralelismo:
  - Potencial de paralelismo massivo;
  - É possível explorar o paralelismo para acelerar certos tipos de comunicação.
- Variedade de aplicações:
  - Processamento digital de sinais (DSP);
  - Redes de computadores;
  - Sistemas embarcados;
  - Computação de alto desempenho (HPC).

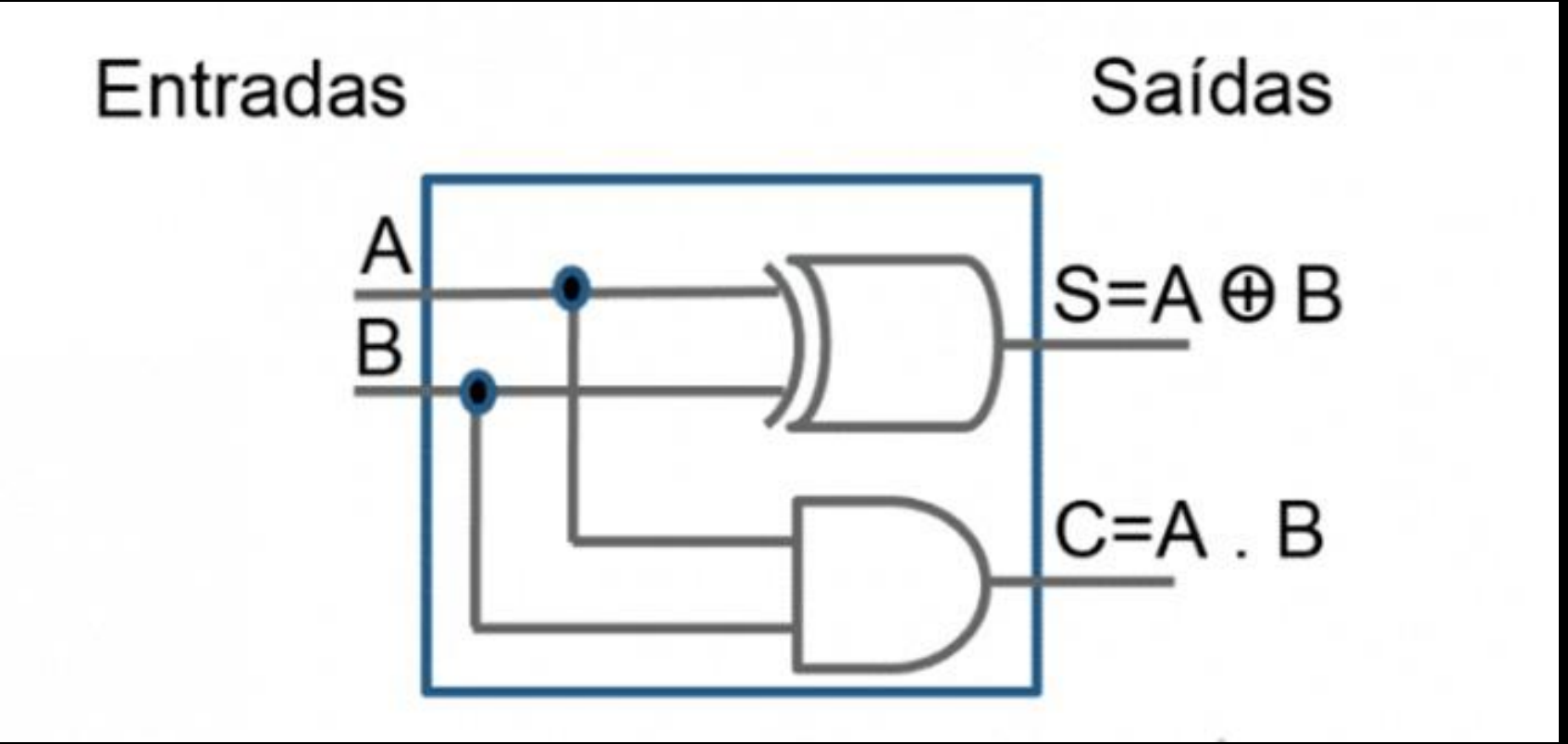
- Unidade de controle (UC):
  - Coordena todas as operações do processador;
  - Decodifica as instruções do programa;
  - Gera os sinais de controle necessários para dirigir as operações de outros componentes do processador.
    - Os sinais são mandados para diferentes partes do processador, para garantir que as operações ocorram na ordem correta.
- Unidade lógica aritmética (ALU):
  - Responsável por realizar operações aritméticas;
  - Operações lógicas;
  - Recebe os operandos do registro;
  - Executa operações conforme instruído pela unidade de controle (UC).

- Registros:
  - Pequenas áreas de armazenamento dentro do processador
  - Armazenamento temporário dos dados que estão sendo processados pela ALU ou UC;
  - Diferentes tipos de registro:
    - Registradores de dados: Armazenam os dados que estão sendo processados.
    - Registradores de endereço: Armazenam endereços de memória para acessar dados.
    - Registrador de instrução: Armazena a instrução atual sendo executada.
- Barramentos:
  - Vias de comunicação;
  - Transferência de dados e sinais de controle entre os componentes do processador, as memórias e outros dispositivos de entrada/saída;
  - Barramento de dados: Transfere dados entre o processador e a memória.
  - Barramento de endereço: Transfere endereços de memória para acessar dados.
  - Barramento de controle: Transfere sinais de controle entre os componentes do processador.

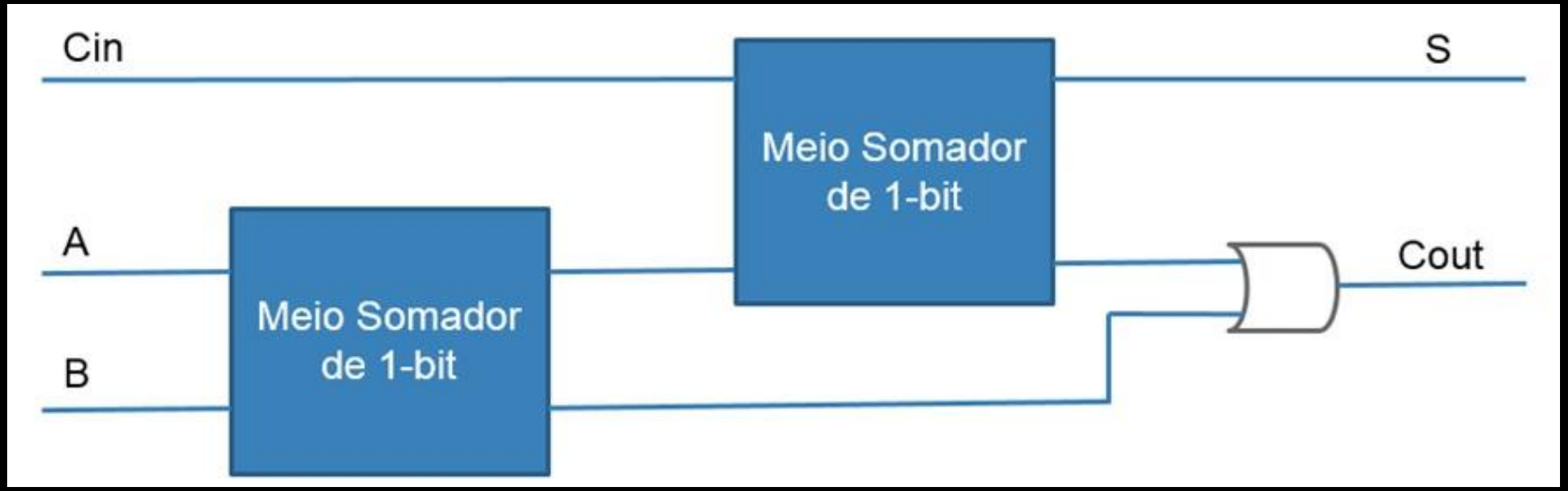
Esses componentes fundamentais do processador estão interconectados de maneira intricada através dos barramentos e sinais de controle gerados pela Unidade de Controle.

- Carry:
  - Conceito da aritmética binária para a soma:
  - Quando há uma soma, cada bit pode carregar um carry ("vai-um") para a próxima posição. Para isso, é necessário uma programação em VDHL
- Carry-in:
  - Bit adicional fornecido como entrada;
  - Permite que a operação ocorra com mais números e carregue um "carry" gerado por uma operação anterior.
- Carry-out:
  - O bit que é "transportado para a parte de fora";
  - Unidade que indica que há uma unidade presente além dos bits de saída;
  - Deve ser adicionada no próximo cálculo.

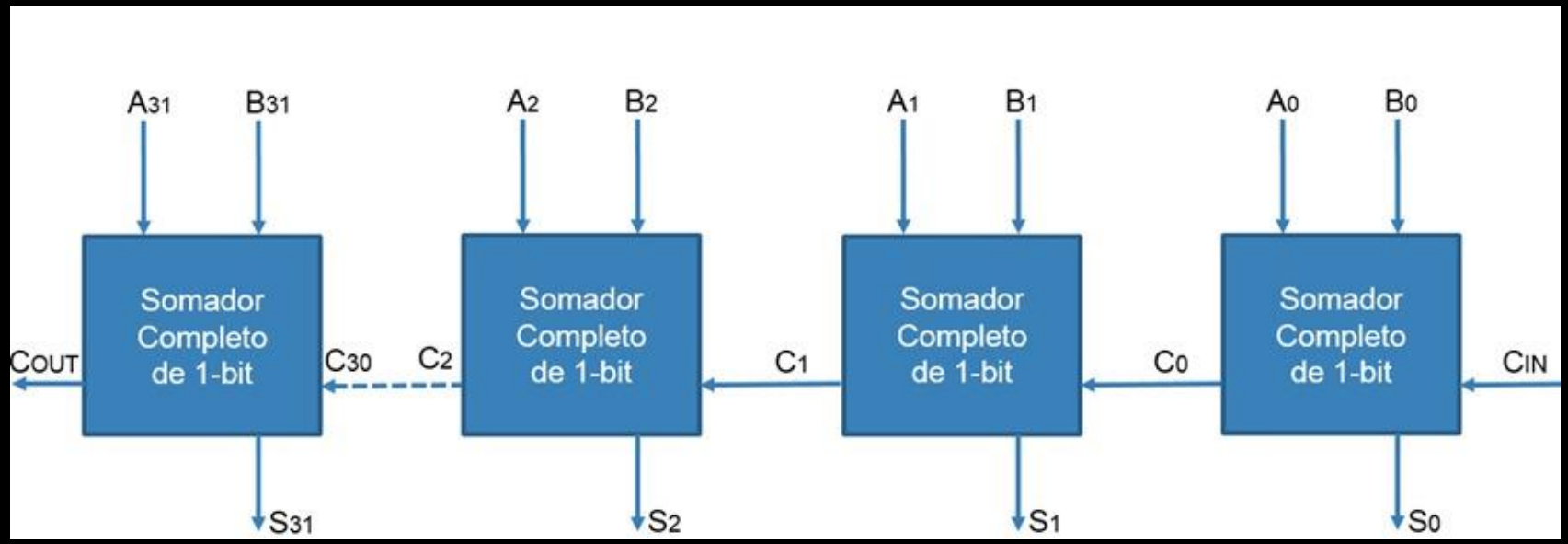
Meio Somador de 1 bit:



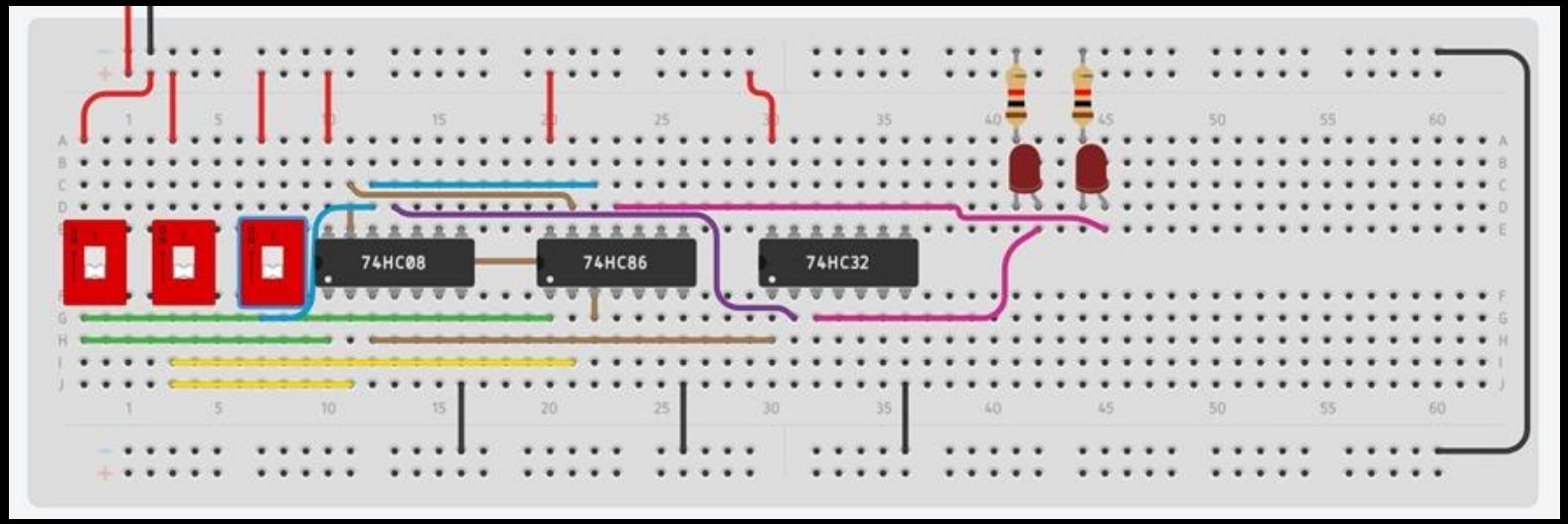
Somador Completo de 1 bit:



Somador Completo de 4 bit:



Circuito Simples Protoboard

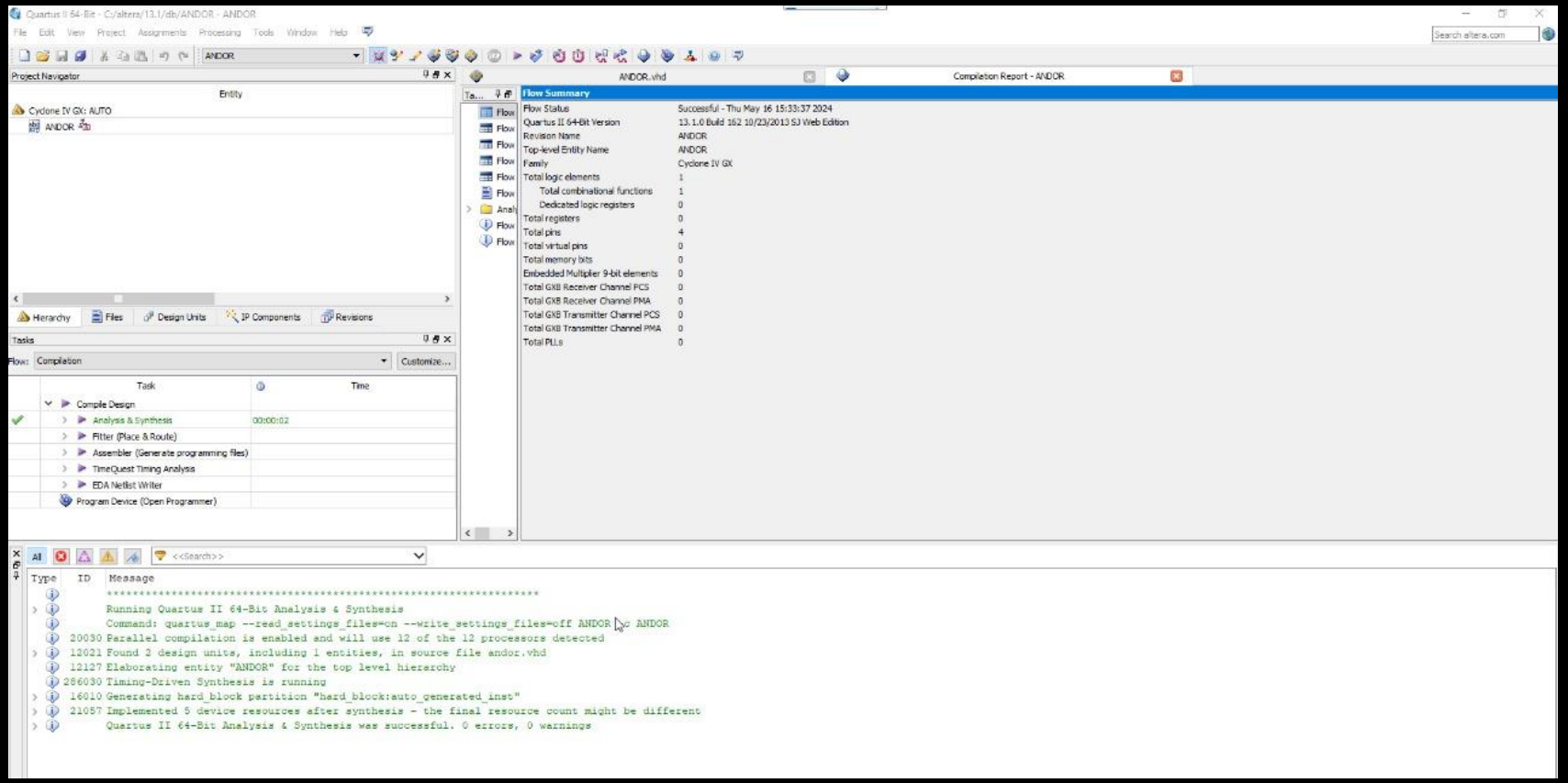


https://www.tinkercad.com/things/bQg4ZwtHJBK-somador-2?sharecode=Aq1R3jnNNP998-7hGDf89ofdEHogQi4_yXPEz0vCV9A

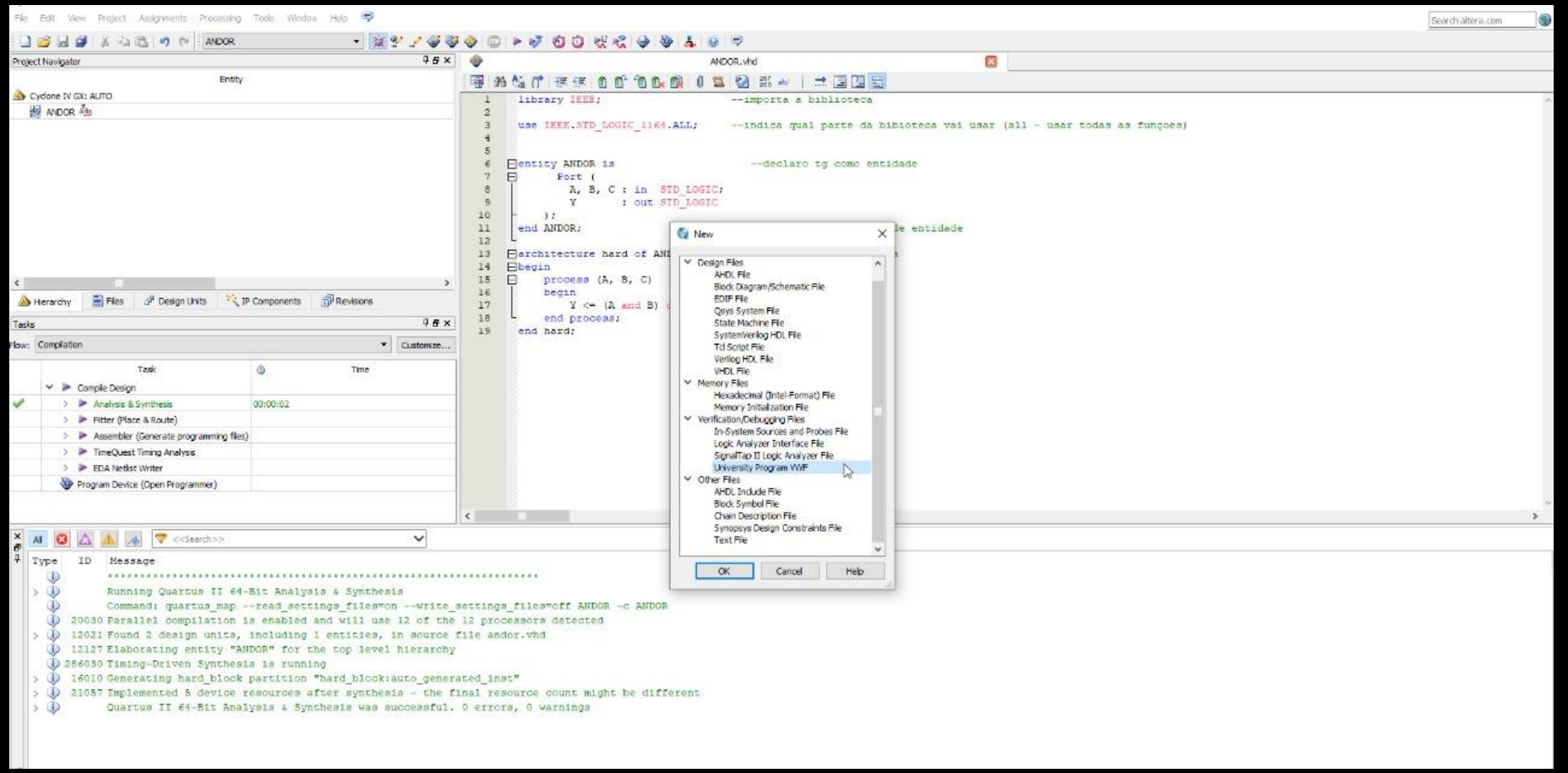
Tabela Verdade do Somador Completo de 1 Bit

A	B	Carry	C-out	Sum	Decimal
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	1	0	2
1	0	0	0	1	1
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	1	1	3

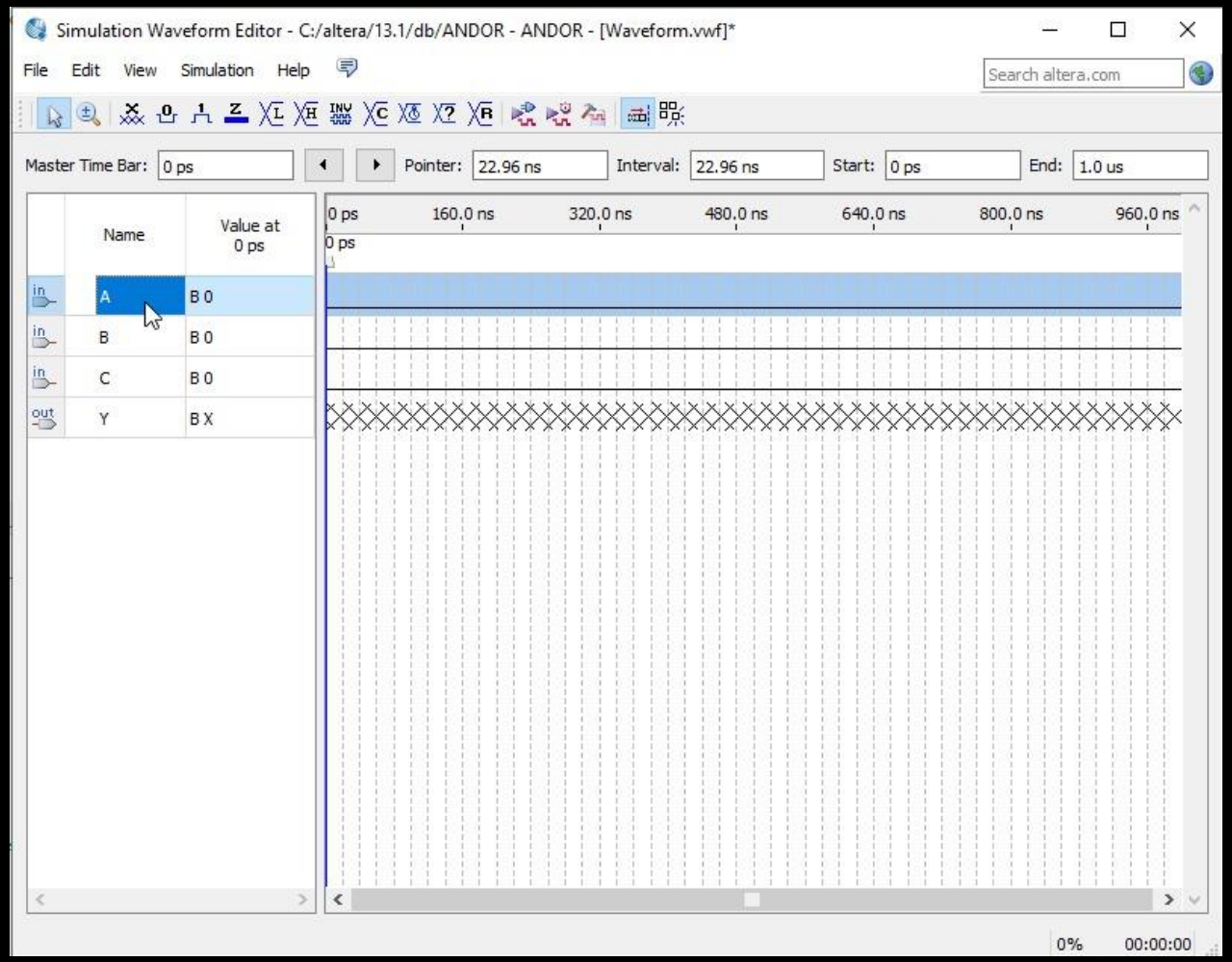
Quartus


Quartus II 64-Bit - C:/altera/13.1/db/ANDOR - ANDOR
Project Navigator
Entity
Cyclone IV GX: AUTO
ANDOR
Hierarchy
Files
Design Units
IP Components
Revisions
Tasks
Flow: Compilation
Customize...
Task
Time
Compile Design
Analysis & Synthesis
00:00:02
Fitter (Place & Route)
Assembler (Generate programming files)
TimeQuest Timing Analysis
EDA Netlist Writer
Program Device (Open Programmer)
ANDOR.vhd
Completion Report - ANDOR
Flow Summary
Flow Status
Successful - Thu May 16 15:33:37 2024
Quartus II 64-Bit Version
13.1.0 Build 162 10/23/2013 SJ Web Edition
Revision Name
ANDOR
Top-level Entity Name
ANDOR
Family
Cyclone IV GX
Total logic elements
Total combinational functions
Dedicated logic registers
Total registers
Total pins
Total virtual pins
Total memory bits
Embedded Multiplier 9-bit elements
Total GXB Receiver Channel PCS
Total GXB Receiver Channel PMA
Total GXB Transmitter Channel PCS
Total GXB Transmitter Channel PMA
Total PLLs
Type
ID
Message
Running Quartus II 64-Bit Analysis & Synthesis
Command: quartus_map --read_settings_files=on --write_settings_files=off ANDOR -c ANDOR
20030 Parallel compilation is enabled and will use 12 of the 12 processors detected
12021 Found 2 design units, including 1 entities, in source file andor.vhd
12127 Elaborating entity "ANDOR" for the top level hierarchy
286030 Timing-Driven Synthesis is running
16010 Generating hard_block partition "hard_block:auto_generated_inst"
21057 Implemented 5 device resources after synthesis - the final resource count might be different
Quartus II 64-Bit Analysis & Synthesis was successful. 0 errors, 0 warnings

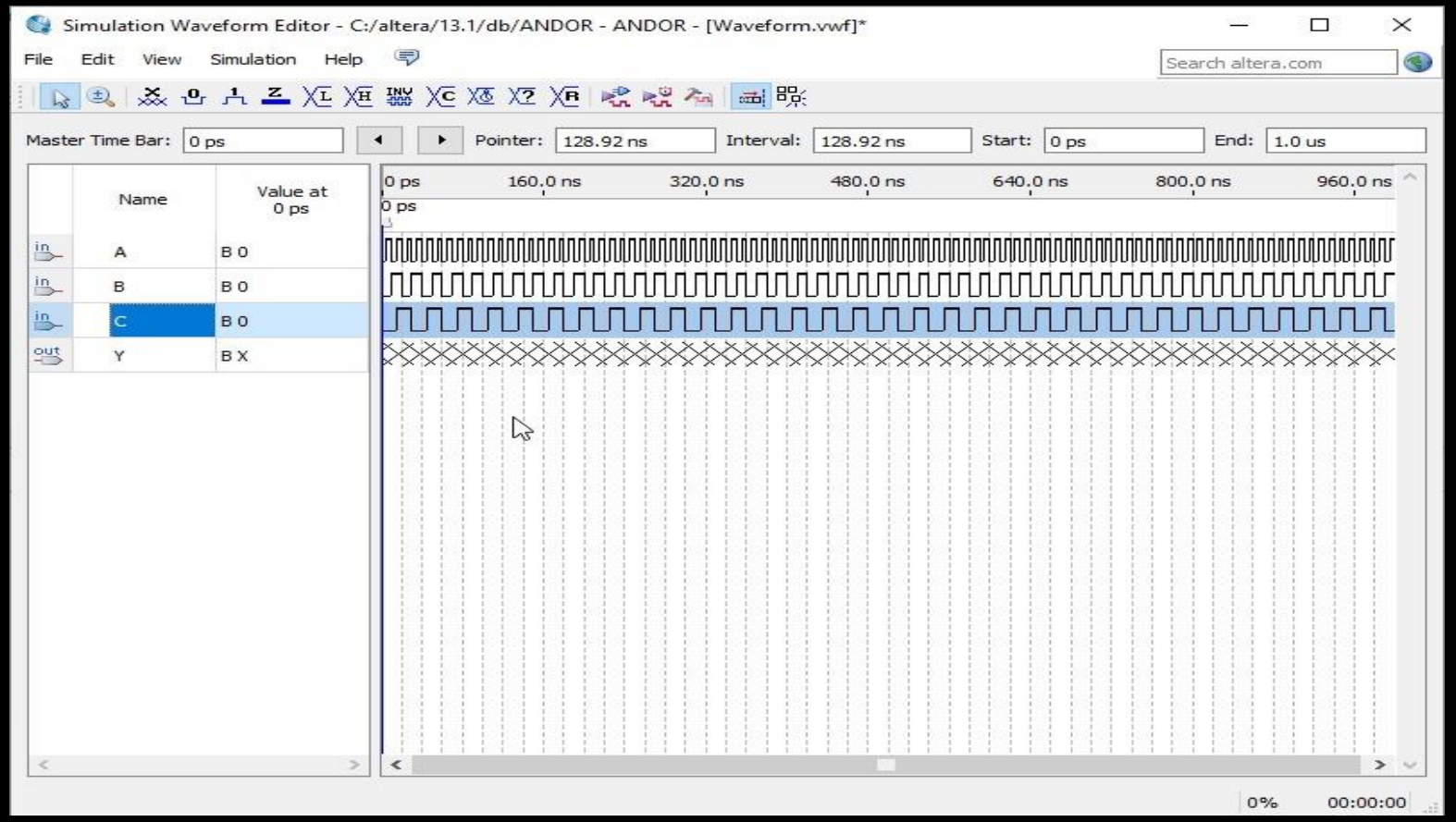
Quartus


File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Entity
Cyclone IV GX: AUTO
ANDOR
Hierarchy Files Design Units IP Components Revisions
Tasks
Flow: Compilation
Customize...
Task Time
Compile Design
Analysis & Synthesis 00:00:02
Fitter (Place & Route)
Assembler (Generate programming files)
TimeQuest Timing Analysis
EDA Netlist Writer
Program Device (Open Programmer)
ANDOR.vhd
1 library IEEE; --importa a biblioteca
2
3 use IEEE.STD_LOGIC_1164.ALL; --indica qual parte da biblioteca vai usar (all - usar todas as funçoes)
4
5
6 entity ANDOR is --declaro tg como entidade
7 Port (
8 A, B, C : in STD_LOGIC;
9 Y : out STD_LOGIC
10);
11 end ANDOR;
12
13 architecture hard of AN
14 begin
15 procesa (A, B, C)
16 begin
17 Y <= (A and B)
18 end procesa;
19 end hard;
New
Design Files
AHDL File
Block Diagram/Schematic File
EDIF File
Qsys System File
State Machine File
SystemVerilog HDL File
Tcl Script File
Verilog HDL File
VHDL File
Memory Files
Hexadecimal (Intel-Format) File
Memory Initialization File
Verification/Debugging Files
In-System Sources and Probes File
Logic Analyzer Interface File
SignalTap II Logic Analyzer File
University Program VWF
Other Files
AHDL Include File
Block Symbol File
Chain Description File
Synopsys Design Constraints File
Text File
OK Cancel Help
All
Type ID Message
Running Quartus II 64-Bit Analysis & Synthesis
Command: quartus_map --read_settings_files=on --write_settings_files=off ANDOR -c ANDOR
20030 Parallel compilation is enabled and will use 12 of the 12 processors detected
12021 Found 2 design units, including 1 entities, in source file andor.vhd
12127 Elaborating entity "ANDOR" for the top level hierarchy
286030 Timing-Driven Synthesis is running
16010 Generating hard_block partition "hard_block:auto_generated_inst"
21057 Implemented 5 device resources after synthesis - the final resource count might be different
Quartus II 64-Bit Analysis & Synthesis was successful. 0 errors, 0 warnings

Quartus

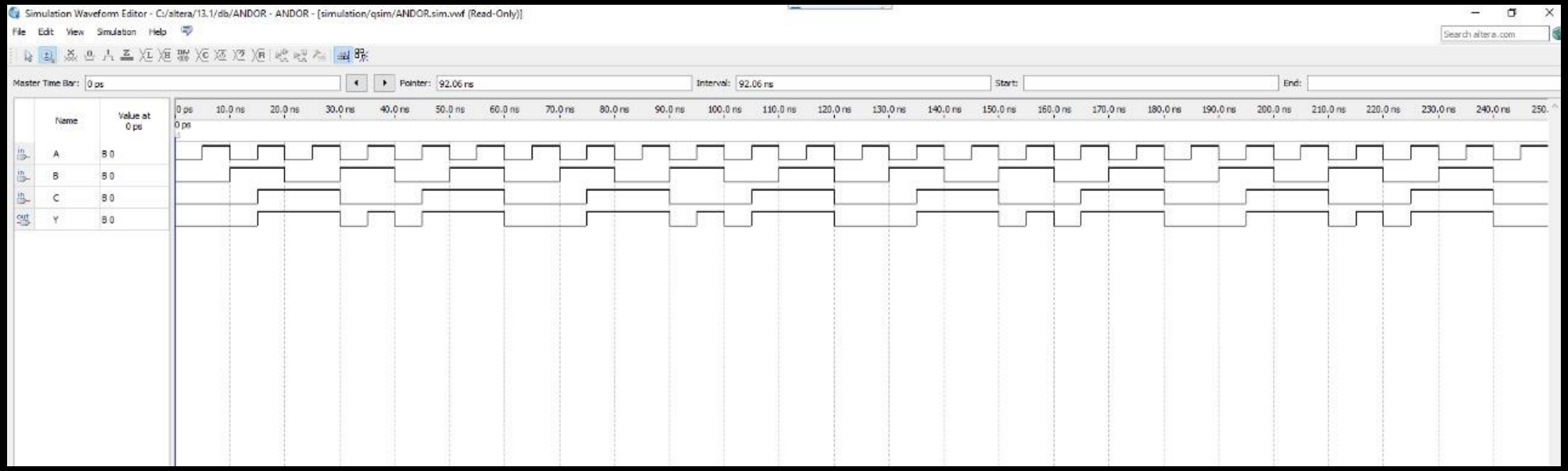

Simulation Waveform Editor - C:/altera/13.1/db/ANDOR - ANDOR - [Waveform.vwf]*
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 22.96 ns
Interval: 22.96 ns
Start: 0 ps
End: 1.0 us
Name
Value at 0 ps
A B 0
B B 0
C B 0
Y B X
0 ps
160.0 ns
320.0 ns
480.0 ns
640.0 ns
800.0 ns
960.0 ns
0%
00:00:00

Quartus



A IN = 1, 10.0 CLOCKS
B IN = 0, 20.0 CLOCKS
C IN = 1, 30.0 CLOCKS

Quartus


Simulation Waveform Editor - C:/altera/13.1/db/ANDOR - ANDOR - [simulation/qsim/ANDOR.sim.vwf (Read-Only)]
File Edit View Simulation Help
Master Time Bar: 0 ps
Pointer: 92.06 ns
Interval: 92.06 ns
Start:
End:
Name
Value at 0 ps
A
B
C
Y
B 0

Quartus

VDHL somador de 1 bit com carry-in

```
library IEEE; --importa a biblioteca


use IEEE.STD_LOGIC_1164.ALL;     --indica qual parte da bibioteca vai
usar (all - usar todas as funçoes)

entity ANDOR is --declaro ANDOR como entidade
Port (
      A, B, C : in  STD_LOGIC;
      Y       : out STD_LOGIC
   );
end ANDOR; --encerra a declarçao de entidade

architecture hard of ANDOR is --inicio da arquitetura
begin
   process (A, B, C)
   begin
      Y <= (A and B) or C;
   end process;
end hard; --fim da arquitetura
```

VDHL somador de 1 bit com carry-in

```
library IEEE; --importa a biblioteca


use IEEE.STD_LOGIC_1164.ALL;     --indica qual parte da bibioteca vai usar (all - usar todas as funçoes)


entity tg is --declaro tg como entidade
Port (
      A, B, C : in  STD_LOGIC;
      Y       : out STD_LOGIC
   );
end tg; --encerra a declarçao de entidade


architecture hard of tg is --inicio da arquitetura
begin
   process (A, B, C)
   begin
      Y <= (A and B) or C;
   end process;
end hard; --fim da arquitetura
```

VDHL somador 4 bits com carry-in

```
library IEEE;
use ieee.std_logic_1164.all;


entity soma_ccarry is
   port (
      a      : in  std_logic_vector(3 downto 0);
      b      : in  std_logic_vector(3 downto 0);
      c_in   : in  std_logic;
      c_out  : out std_logic;
      s      : out std_logic_vector(3 downto 0)
   );
end soma_ccarry;


architecture hardware of soma_ccarry is
begin
   process(a, b, c_in)
   begin
      for i in 0 to 3 loop
         s(i) <= a(i) XOR b(i) XOR c_in;
         if i = 0 then
            c_out <= (a(i) AND b(i)) or (a(i) AND c_in) or (b(i) AND c_in);
         else
            c_out <= (s(i-1) AND ((a(i) XOR b(i)) or (a(i) XOR c_in) or (b(i) XOR c_in))) or (a(i) AND b(i) AND
c_in);
         end if;
      end loop;
   end process;
end hardware;
```