A logo with blue and green squares

Description automatically generatedUniversidade de Brasília – UNB

Curso: Engenharia de Redes de Comunicação  
Disciplina: Laboratório de Sistemas Digitais  
Turma: 08

Relatório da Disciplina Laboratório de Sistemas Digitais  
Tema: Experimento 02 – Introdução à linguagem VHDL

Aluno: Pedro Henrique Dias Avelar  
Matrícula: 241037112  
Professor: Eduardo Paiva

# Introdução

O presente experimento tem os seguintes objetivos:

• Implementar circuitos combinacionais simples utilizando a linguagem VHDL.

• Desenvolver módulos básicos – somador completo e multiplexador – que podem ser usados

futuramente para implementar circuitos mais complexos.

• Simulação no ModelSim.

# Experimento 01 – Somador Completo

O somador completo é descrito pelas funções lógicas abaixo:

Podemos representar as funções acima no Logisim pelo código abaixo:

-- Experimento 02 - Questão 01

-- Aluno: Pedro Henrique Dias Avelar 241037112

-- Turma 08

-- Data: 02/11/2024

-- Funções lógicas do somador completo:

-- S = A xor B xor Cin

-- Cout = AB or ACin or BCin

**LIBRARY** IEEE**;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL;**

**ENTITY** CIRCUITO\_E02Q01 **IS**

**PORT** **(**A**,**B**,**CIN**:** **IN** STD\_LOGIC**;** --ENTRADAS

S**,**COUT**:** **OUT** STD\_LOGIC**);** --SAÍDAS

**END** CIRCUITO\_E02Q01**;**

**ARCHITECTURE** ARC\_CIRCUITO\_E02Q01 **OF** CIRCUITO\_E02Q01 **IS**

**BEGIN**

S **<=** A **XOR** B **XOR** CIN**;** --S = A xor B xor Cin

COUT **<=** **(**A **AND** B**)** **OR** **(**A **AND** CIN**)** **OR** **(**B **AND** CIN**);** --Cout = AB or ACin or BCin

**END** ARC\_CIRCUITO\_E02Q01**;**

Para a simulação do circuito, foram realizados os seguintes procedimentos:

A screenshot of a computer

Description automatically generatedInicialmente a compilação do código VHDL;

A screenshot of a computer

Description automatically generatedEm seguida selecionei a opção “Start Simulation...” na aba “Simulate”

A screenshot of a computer program

Description automatically generated Para a simulação, selecionei a arquitetura ARC\_CIRCUITO\_E02Q01 descrita no código VHDL acima

A screenshot of a computer

Description automatically generated Repassei as portas lógicas de entrada e saída da aba “Objects” para a aba “Wave

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

O sinal das portas lógicas A, B e Cin foi configurado “Clock” com a opção “First Edge – Falling” e um período de 40 para a porta A, 20 para a porta B e 10 para porta C. Isto foi feito para que a forma de onda da simulação se aproximasse da tabela verdade.

A screenshot of a computer program

Description automatically generated A “Run Length foi ajustada para 80, para que a simulação então envolvesse apenas todas as combinações de entrada para as portas A,B e Cin

A screenshot of a computer

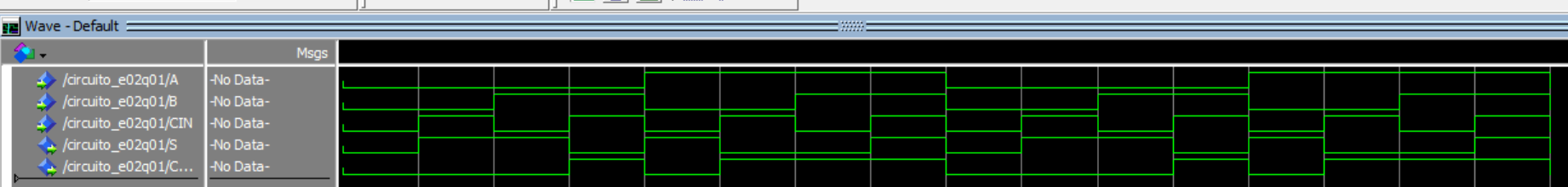
Description automatically generated Executei então a simulação com a opção “Simulate->Run->Run 100”

A screen shot of a computer

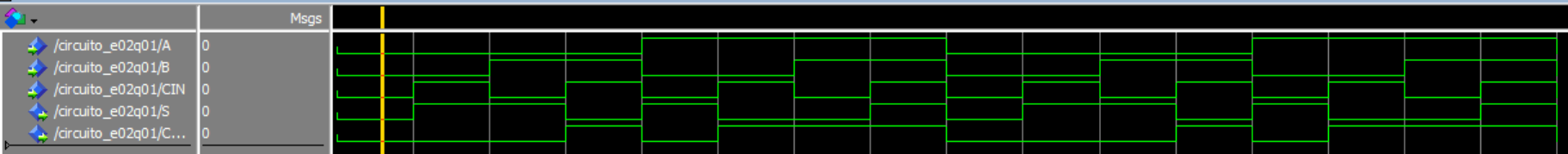
Description automatically generated Este foi o resultado da simulação;

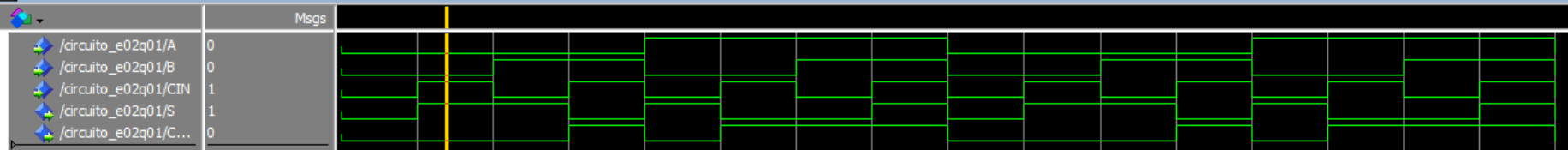
A screenshot of a computer

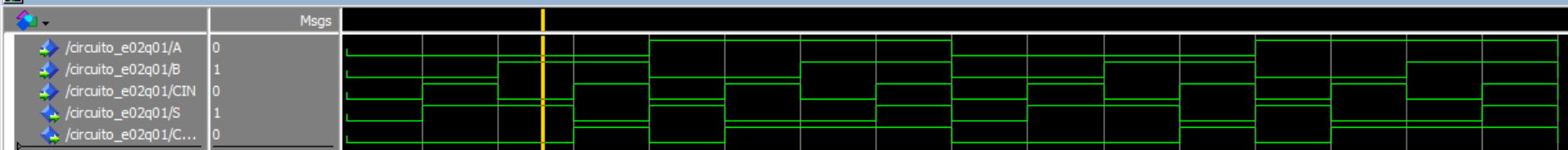
Description automatically generatedPorém o resultado pode ser melhor visualizado com a opção “Zoom Full”

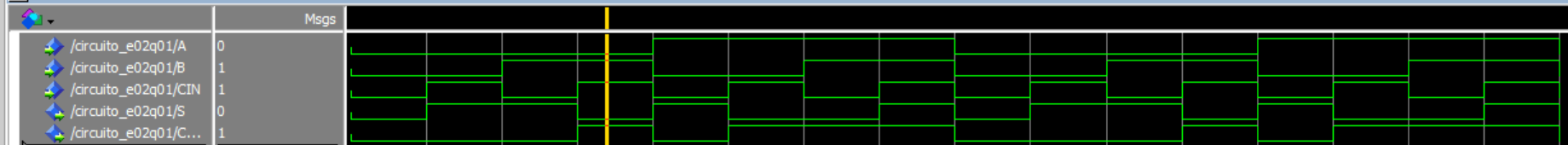


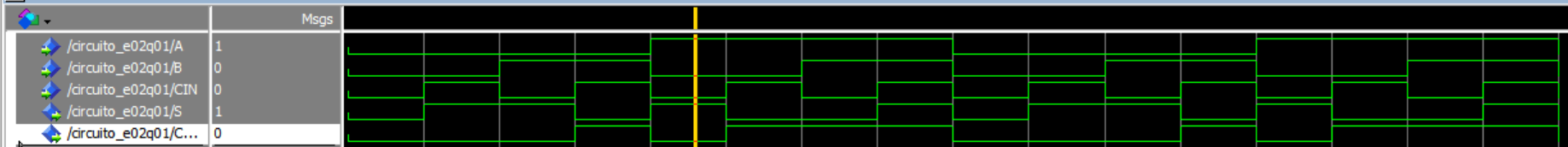
O sinal das entradas e saídas pode ser obtido clicando no gráfico de onda – Na coluna “Msgs” temos o sinal da respectiva porta para o período (linha amarela) selecionado:

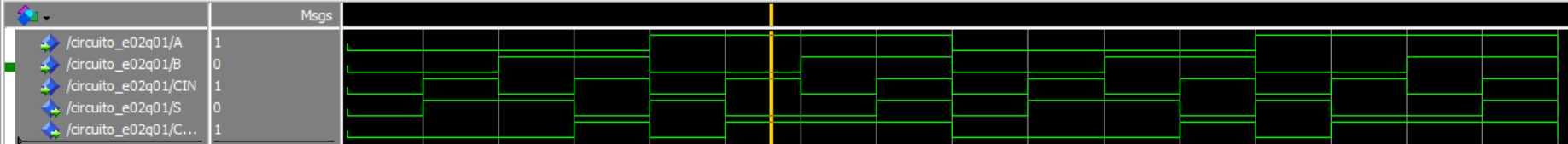


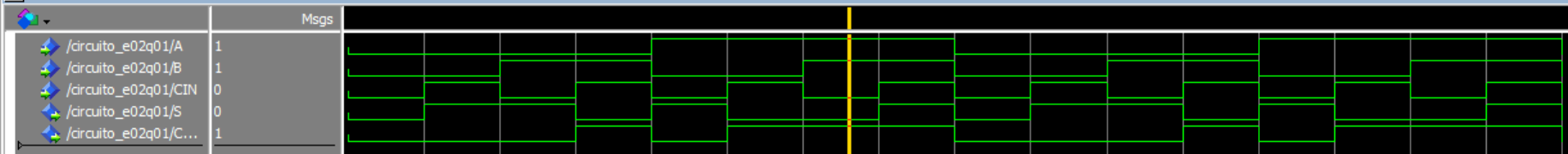


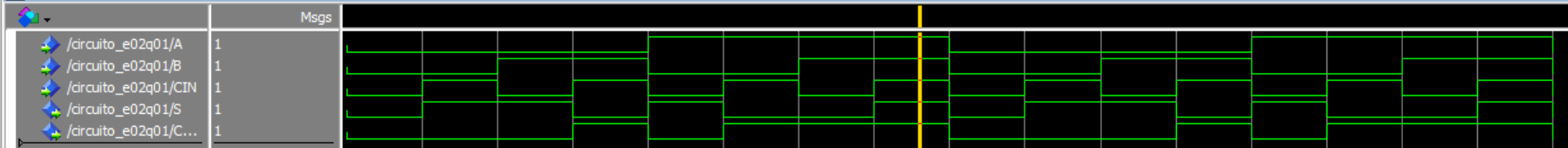












Montando a tabela-verdade, temos então que:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | Cin | S | Cout |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

# Experimento 02 – Multiplexador de 4 para 1

O multiplexador de 4 para 1 é descrito pela função lógica abaixo:

O multiplexador foi representado no Logisim pelo código abaixo:

-- Experimento 02 - Questão 02

-- Aluno: Pedro Henrique Dias Avelar 241037112

-- Turma 08

-- Data: 02/11/2024

-- Função lógica do multiplexador 4 para 1:

-- Y = D0!S1!S0 + D1!S1S0 + D2S1!S0 + D3S1S0

**LIBRARY** IEEE**;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL;**

**ENTITY** CIRCUITO\_E02Q02 **IS**

**PORT** **(**D**:** **IN** STD\_LOGIC\_VECTOR **(**3 **DOWNTO** 0**);** --ENTRADA

S**:** **IN** STD\_LOGIC\_VECTOR **(**1 **DOWNTO** 0**);** --SELEÇÃO

Y**:** **OUT** STD\_LOGIC**);** --SAÍDA

**END** CIRCUITO\_E02Q02**;**

**ARCHITECTURE** ARC\_CIRCUITO\_E02Q02 **OF** CIRCUITO\_E02Q02 **IS**

**BEGIN**

Y **<=** **(**D**(**0**)** **AND** **NOT(**S**(**1**))** **AND** **NOT(**S**(**0**)))** **OR** --D0!S1!S0

**(**D**(**1**)** **AND** **NOT(**S**(**1**))** **AND** S**(**0**))** **OR** --D1!S1S0

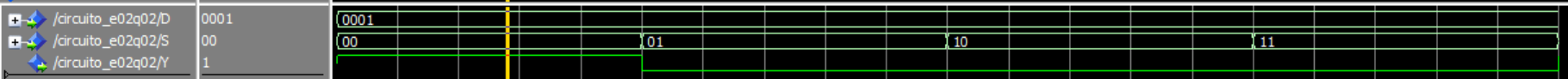
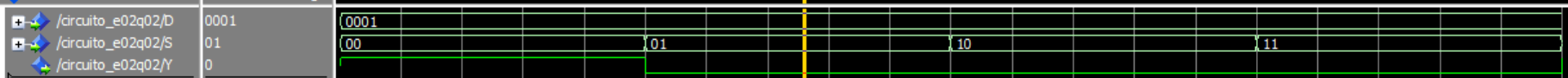
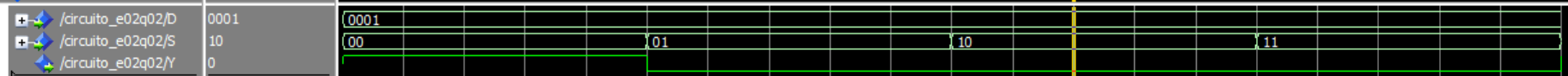
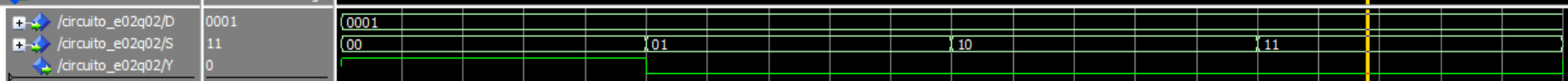
**(**D**(**2**)** **AND** S**(**1**)** **AND** **NOT(**S**(**0**)))** **OR** --D2S1!S0

**(**D**(**3**)** **AND** S**(**1**)** **AND** S**(**0**));** --D3S1S0

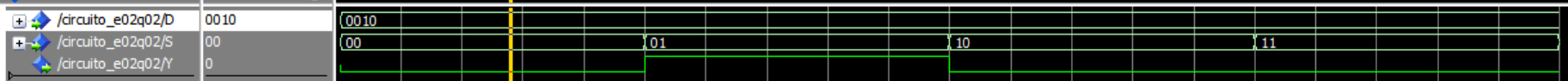
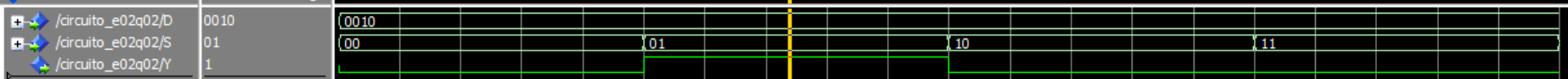
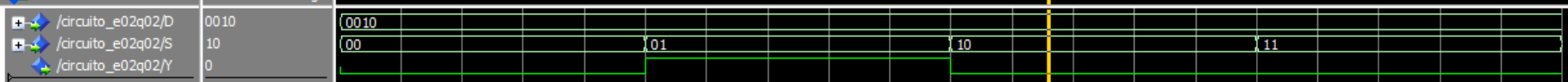
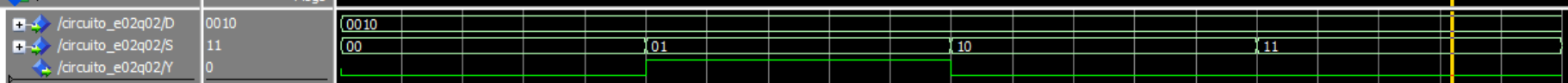
**END** ARC\_CIRCUITO\_E02Q02**;**

E a simulação foi então realizada para D = 0001, 0010, 0100 e 1000 e a seleção alternando entre 00, 01, 10 e 11:

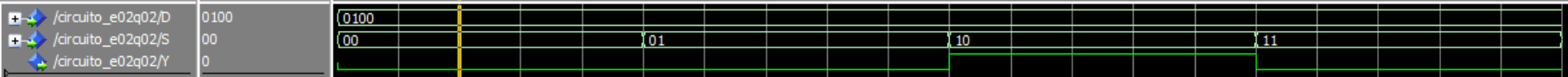
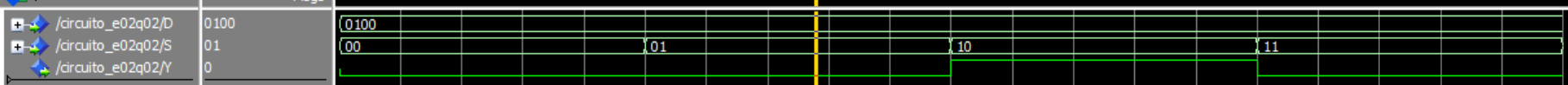
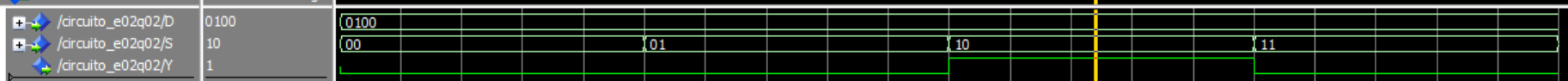
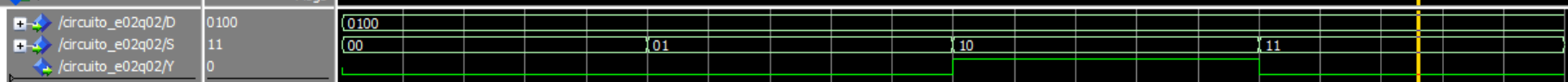
D=0001

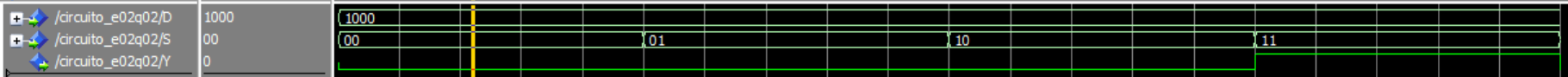
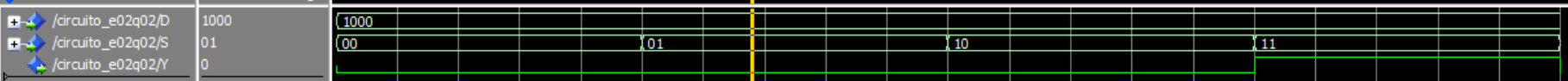
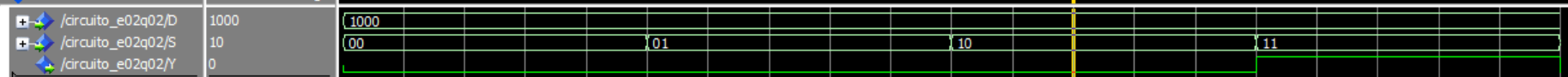
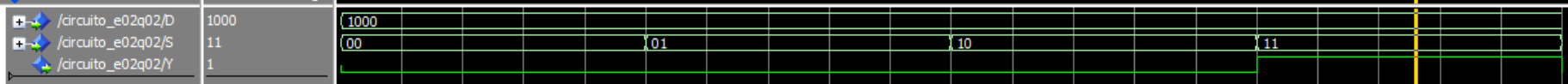
D=0010

D=0100

D=1000

Montando a tabela-verdade, temos então que:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| D3 | D2 | D1 | D0 | S1 | S0 | Y |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 |