UNIVERSIDADE FEDERAL DE SANTA CATARINA

CENTRO TECNOLÓGICO

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

CURSO DE CIÊNCIAS DA COMPUTAÇÃO

INE5406 - SISTEMAS DIGITAIS

PROJETO PRÁTICO DE SISTEMAS DIGITAIS:

UNIDADE PARA CÁLCULO DO MÁXIMO DIVISOR COMUM DE DOIS NÚMEROS INTEIROS POSITIVOS

Equipe:

Caetano Colin Torres

Florianópolis

Setembro, 2018

Sumário

Sumário

1. Introdução.

Este projeto prático de sistemas digitais tem o fim de implementar um sistema digital síncrono que extraia o máximo divisor comum de dois números inteiros positivos representados em binário puro, esse cálculo será feito por meio do Algoritmo de Euclides(baseado em subtrações).

2.Projeto do Sistema.

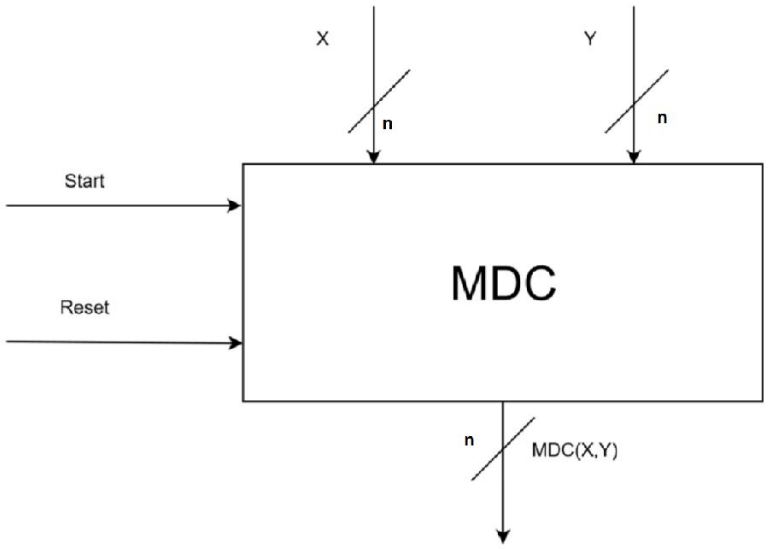
O projeto desse sistema digital inicia-se com o desenvolvimento do algoritmo e a captura da máquina de estados finitos de alto nível.Depois é construído o bloco operacional e o bloco de controle e obtém-se a FSM(Máquina de Estados Finitos).Depois de capturar a FSM e projetar os blocos, o algoritmo é sintetizado em VHDL.

2.1 Identificação das entradas e saídas.

Entradas: Start(1 bit), Reset(1 bit), X(n bits), Y(n bits)

Saídas: MDC(X,Y)(n bits)

n bits serão definidos conforme a capacidade da placa.



2.2 Captura do comportamento.

O máximo divisor comum entre dois números inteiros positivos pode ser obtido a partir do algoritmo de euclides(versão inicial), esta que executa subtrações sucessivas atualizando o valor dos parâmetros recebidos(no caso do MDC(X,Y), esses parâmetros são X e Y).Com isso, atualiza-se o valor deles por meio de subtrações até os valores deles serem iguais, quando isso ocorre, o MDC é o próprio valor final obtido.

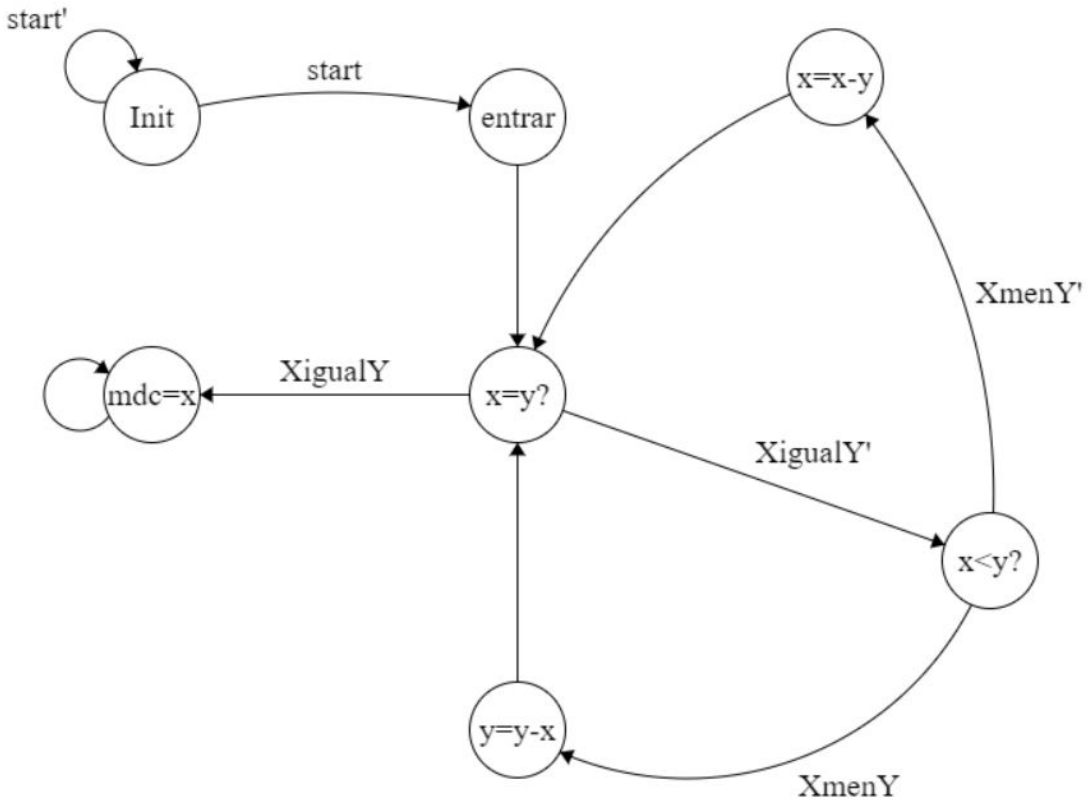
O algoritmo de euclides pode ser escrito em java como:

```
public int mdc(int x,int y) {
            while (x != y) {

                  if (x < y) {
                   y = y - x;
                  }

              else {
                   x = x - y;
              }

            }
            return x;
      }
```

2.3 FSM de alto nível.

Analisando o algoritmo, nota-se que será preciso de estados nos quais serão feitas comparações, como x=y e x<y, essas comparações gerarão sinais de status que sinalizarão a troca de estado.Também constata-se que serão necessários estados em que serão realizadas operações aritméticas, nesse caso serão realizadas subtrações e como serão feitas apenas subtrações de inteiros positivos, será implementado 2 subtratores e unidades comparativas,assim não é preciso se preocupar com overflow nesse projeto.

A partir do algoritmo captura-se a FSM de alto nível:



2.4 Projeto do bloco operativo.

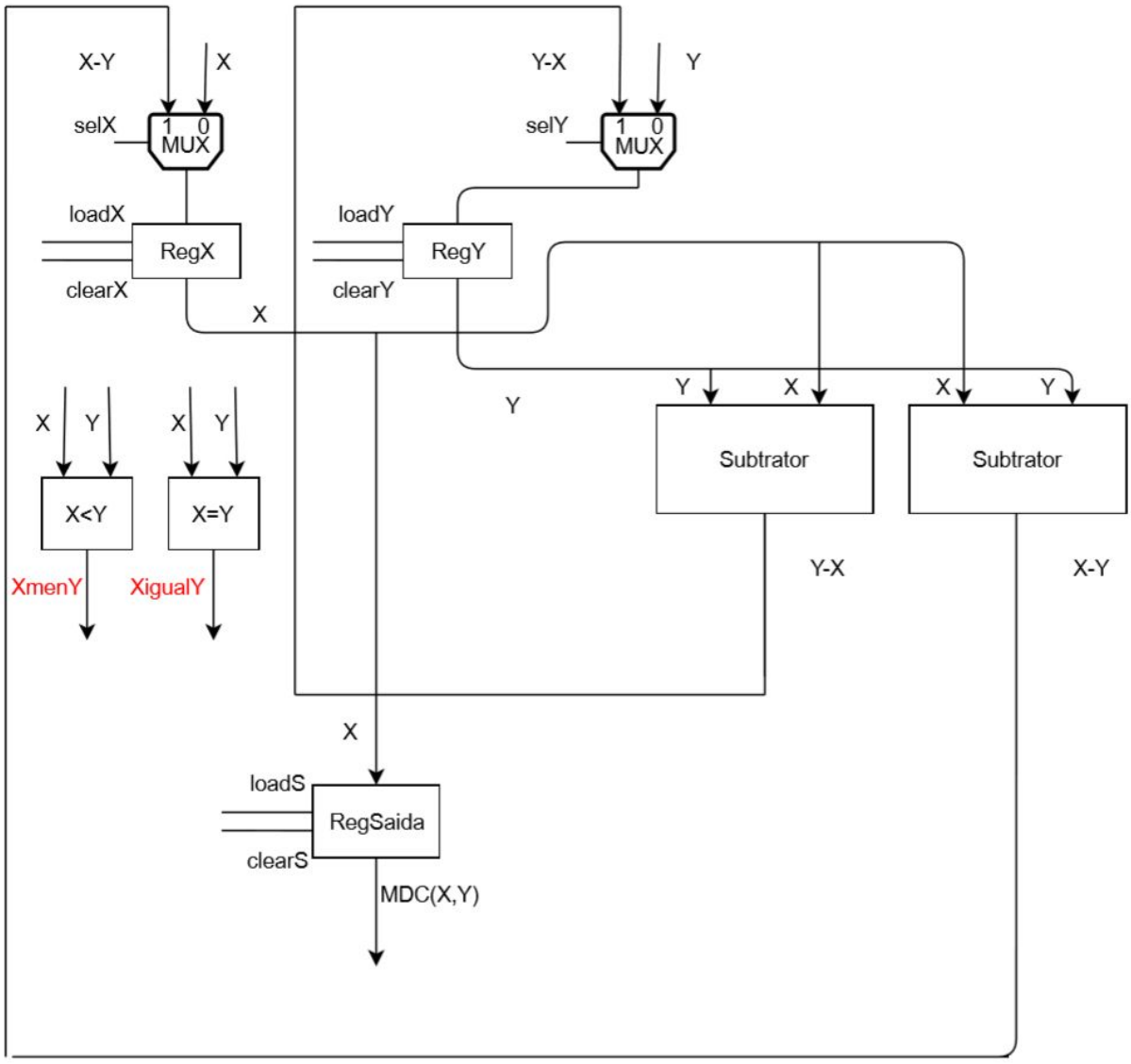
Ao analisar a máquina de estados finitos de alto nível, observa-se que serão necessários 2 registradores, um para X e outro para Y, além disso será preciso de 2 multiplexadores 2:1, cada um desses associado a uma das entradas dos registradores de X e de Y, o sinais de seleção desses multiplexadores serão: selX para X e selY para Y.Serão implementados 2 subtradores, um efetuará a operação X-Y e outro Y-X.Serão necessárias 2 unidades comparativas que gerarão

sinais de status para o bloco de controle, uma sinalizará se X<Y e outra sinalizará se X=Y.Será implementado um registrador que retorna o valor final do máximo divisor comum entre os dois números informados(X,Y) esse registrador será chamado de RegSaida.É necessário reforçar que os sinais X e Y, tanto no bloco X=Y e X<Y são as saídas dos registradores de X e de Y.

A partir da FSM de alto nível projeta-se o bloco operativo:

Entradas: X(inicial),Y(inicial),loadX,loadY,clearX,clearY,selX,selY,loadS,clearS,clock

Saídas:MDC(X,Y),XmenY,XigualY



2.5 Projeto do Bloco de Controle.

O bloco de controle será responsável pelo controle do bloco operativo, ele dirá quando que um determinado registrador deve funcionar ou resetar e qual será o chaveamento dos multiplexadores em determinados estados, além disso ele será responsável pela lógica de próximo estado da máquina de estados finitos.Nesse projeto, este bloco será projetado em VHDL.

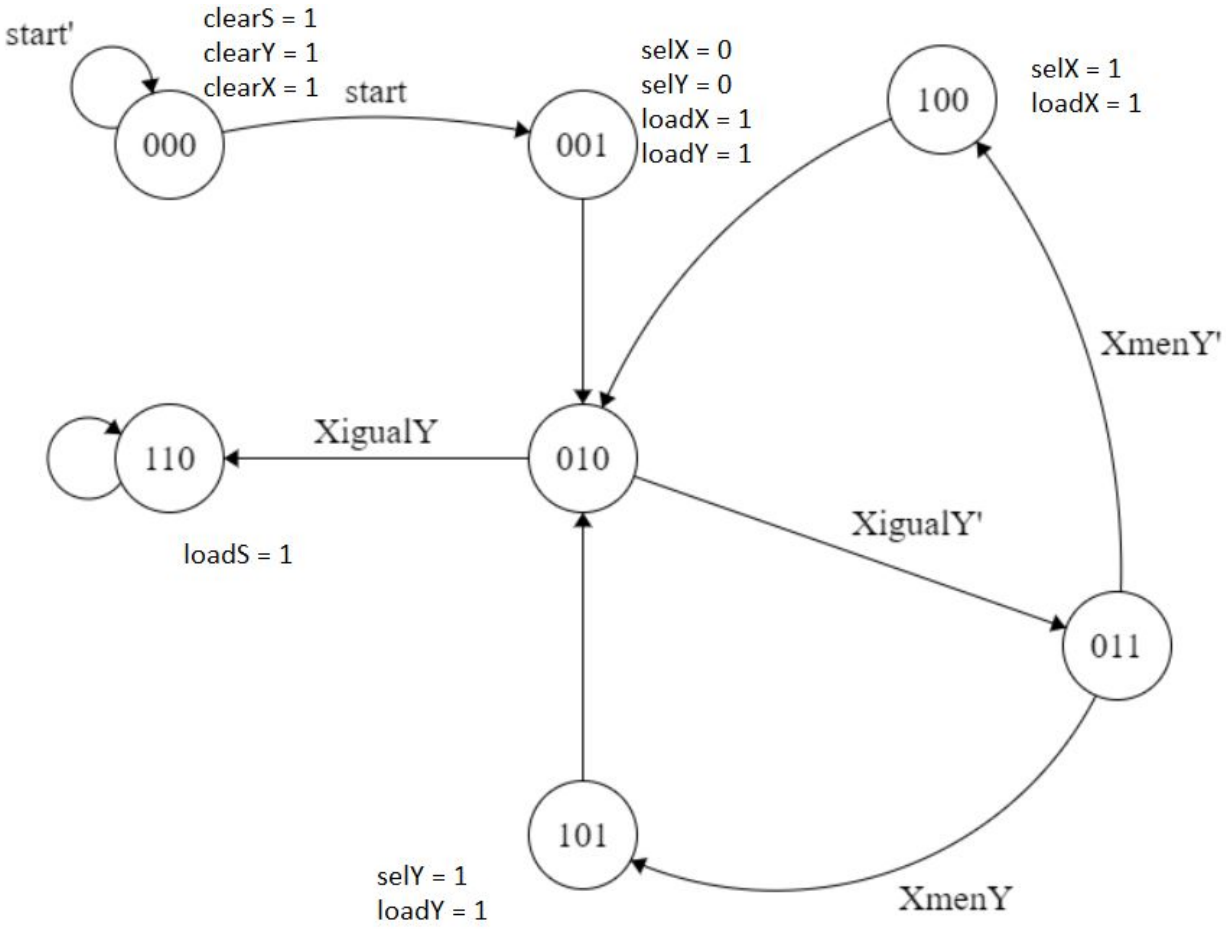
Entradas: start, reset, XmenY,XigualY,clock

Saídas:clearS,clearY,clearX,loadX,loadY,loadS,selX,selY

2.6 FSM de baixo nível.

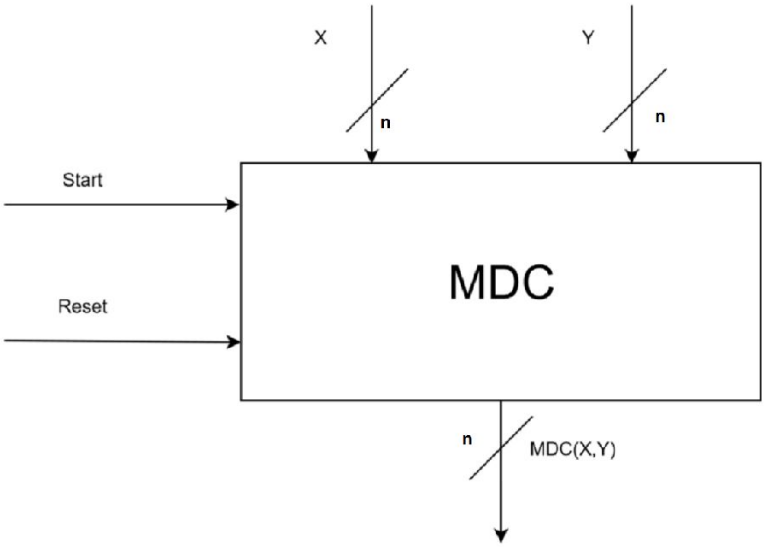
Com o bloco operativo projetado(Datapath), e a FSM de alto nível, pode-se projetar a FSM de baixo nível.Constata-se que no estado 000(Init na máquina de alto nível) será necessário resetar os valores de todos registradores, uma vez que quando o sinal reset é igual a 1, volta-se ao estado 000 e pode-se fazer um novo cálculo de máximo divisor comum com parâmetros diferentes.Quando o sinal start = 1, passa-se para o estado 001, nesse o sinal selX = 0 e selY = 0, estão chaveando o valor de X e de Y inicial para a entrada do registradores de X e de Y, respectivamente.Assim, os sinais loadX e loadY dos registradores serão colocados em nivel alto, e os valores serão registrados com segurança para os próximos estados(um ciclo de clock é garantido).Depois de estar no estado 001 e um ciclo de clock ser executado, o próximo estado da FSM é o 010, nesse será feito a comparação que sinalizará se X=Y, se X for igual a Y, o algoritmo finaliza sua execução e passa para o estado 110, onde o valor da saida é retornado, senão o próximo estado é 011 onde será feita outra comparação, nessa comparação se X for menor do que Y, o valor de Y será atualizado para Y-X, isso ocorre quando é sinalizado que X é menor do que Y e é feita a transição de estado de 011 para 101 onde é feita a operação aritmética Y-X, o registrador de Y é ativado e o sinal de seleção do mux 2:1 de Y é sinalizado como alto.Caso ocorra que X é maior do que Y, o estado vai de 011 para 100 onde o valor de X é atualizado para X-Y, para isso é necessário o sinal selX ser definido como alto, e o sinal de load do registrador de X definido como alto também, assim X atualiza seu valor para X-Y.Quando o estado é 100 ou 101(onde são atualizados os valores dos parâmetros) o próximo estado sempre será o 010(onde é feita a comparação), esse processo é repetido até X ser sinalizado como igual a Y, passando para o estado 110.Vale reforçar que esta máquina de estados é uma máquina de Moore.

A FSM de baixo nível obtida é a seguinte:

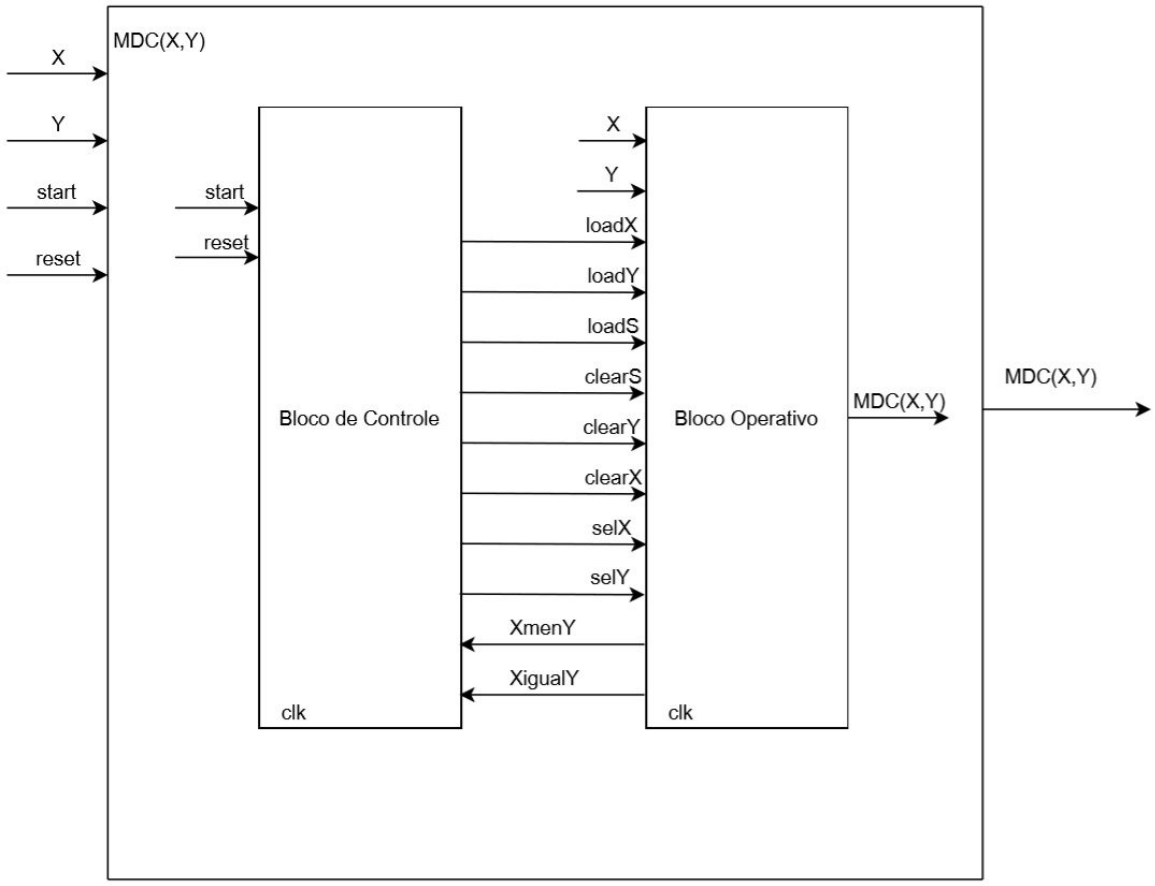


2.7 Integração do bloco de controle e do bloco operativo.

Para a unidade para o cálculo do máximo divisor comum entre dois números ficar da seguinte maneira:



é necessária a integração do bloco de controle com o bloco operativo, que é a seguinte:



3.Desenvolvimento.

3.1 Componentes em Linguagem de Descrição de Hardware.

código VHDL de cada componente desenvolvido, informações sobre sua temporização e área, e esquemático (RTL).

Lembrando que os testes foram parametrizados para 4 bits, ou seja o número de células lógicas utilizadas, informação sobre temporização e o esquemático RTL podem variar de acordo com a quantidade de bits mapeada para o hardware e as informações abaixos condizem com os testes feitos.

Device usado na compilação/simulação: Ciclone II EP2C50F672C8.

3.1.1 Registradores de n bits.

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

entity Register_Nbits is
    generic (N: positive := 4);
    port(
        clk, reset,carga: in std_logic;
        d: in std_logic_vector(N-1 downto 0);
        q: out std_logic_vector(N-1 downto 0)
    );
end entity;

architecture canonic of Register_Nbits is
    subtype InternalState is std_logic_vector(N-1 downto 0); -- ...
    signal nextState, currentState: InternalState;
begin
    -- next state logic (combinatorial)
    nextState <= d;

    -- memory element (sequential)
    ME: process (clk, reset) is
    begin
        if reset='1' then
            currentState <= (others=>'0');
        elsif rising_edge(clk) and carga = '1' then
            currentState <= nextState;
        end if;
    end process;

    -- output logic (combinatorial)
    q <= currentState;

end architecture;
```

Setup Times

	Data Port	Clock Port	Rise	Fall	Clock Edge	Clock Reference
1	d[*]	clk	4.205	4.205	Rise	clk
1	d[2]	clk	4.205	4.205	Rise	clk
2	d[1]	clk	3.858	3.858	Rise	clk
3	d[3]	clk	3.719	3.719	Rise	clk
4	d[0]	clk	-0.533	-0.533	Rise	clk
2	carga	clk	-0.020	-0.020	Rise	clk

Hold Times

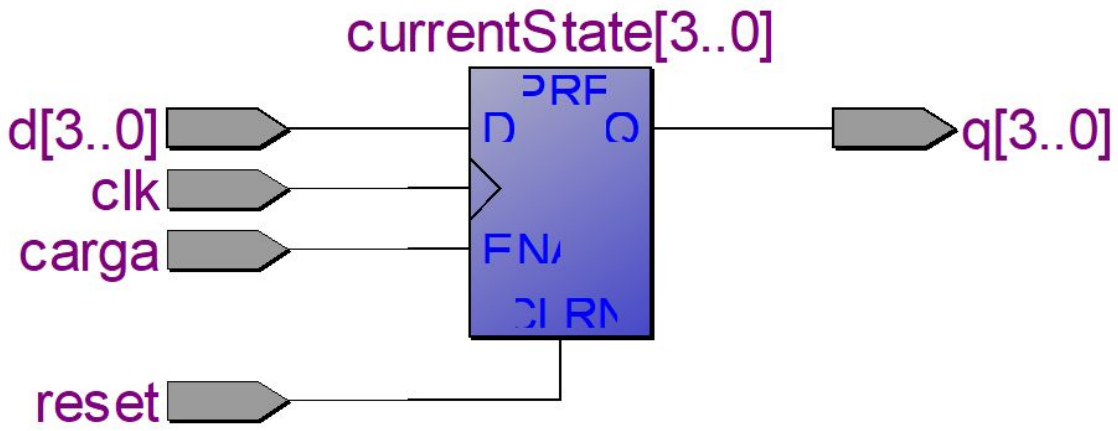
	Data Port	Clock Port	Rise	Fall	Clock Edge	Clock Reference
1	d[*]	clk	0.799	0.799	Rise	clk
1	d[0]	clk	0.799	0.799	Rise	clk
2	d[3]	clk	-3.453	-3.453	Rise	clk
3	d[1]	clk	-3.592	-3.592	Rise	clk
4	d[2]	clk	-3.939	-3.939	Rise	clk
2	carga	clk	0.286	0.286	Rise	clk

Clock to Output Times

	Data Port	Clock Port	Rise	Fall	Clock Edge	Clock Reference
1	q[*]	clk	7.999	7.999	Rise	clk
1	q[0]	clk	7.625	7.625	Rise	clk
2	q[3]	clk	7.637	7.637	Rise	clk
3	q[2]	clk	7.989	7.989	Rise	clk
4	q[1]	clk	7.999	7.999	Rise	clk

Fitter Summary

Fitter Status	Successful - Thu Oct 04 17:27:17 2018
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	UnidadeMDC_Caetano
Top-level Entity Name	Register_Nbits
Family	Cyclone II
Device	EP2C50F672C8
Timing Models	Final
Total logic elements	4 / 50,528 (< 1 %)
Total combinational functions	0 / 50,528 (0 %)
Dedicated logic registers	4 / 50,528 (< 1 %)
Total registers	4
Total pins	11 / 450 (2 %)
Total virtual pins	0
Total memory bits	0 / 594,432 (0 %)
Embedded Multiplier 9-bit elements	0 / 172 (0 %)
Total PLLs	0 / 4 (0 %)



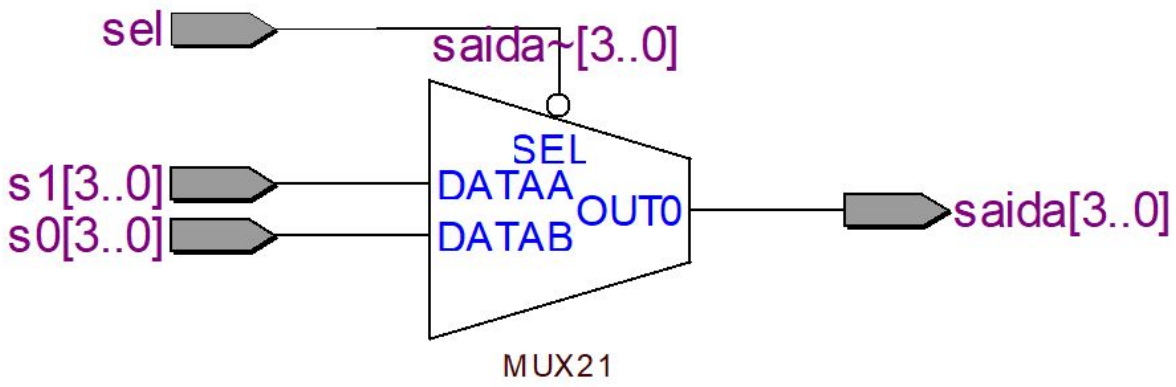
3.1.2 Multiplexadores 2:1 de n bits.

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity muxNbits is
    generic (n: natural := 4);
    port (
s0,s1: in std_logic_vector (n-1 downto 0) ;
sel : in std_logic;
saida: out std_logic_vector(n-1 downto 0)
);
end muxNbits;

architecture arq of muxNbits is
begin
saida <= s0 when sel = '0' else s1;
end arq;
```

Propagation Delay

	Input Port	Output Port	RR	RF	FR	FF
1	s0[1]	saida[1]	12.069			12.069
2	sel	saida[0]	12.030	12.030	12.030	12.030
3	s1[2]	saida[2]	11.797			11.797
4	sel	saida[2]	11.736	11.736	11.736	11.736
5	s0[2]	saida[2]	11.735			11.735
6	s1[1]	saida[1]	11.414			11.414
7	s1[3]	saida[3]	11.411			11.411
8	sel	saida[3]	11.376	11.376	11.376	11.376
9	sel	saida[1]	11.365	11.365	11.365	11.365
10	s0[3]	saida[3]	11.254			11.254
11	s0[0]	saida[0]	7.540			7.540
12	s1[0]	saida[0]	7.352			7.352

Flow Summary	
Flow Status	Successful - Thu Oct 04 17:44:28 2018
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	UnidadeMDC_Caetano
Top-level Entity Name	muxNbits
Family	Cyclone II
Device	EP2C50F672C8
Timing Models	Final
Total logic elements	4 / 50,528 (< 1 %)
Total combinational functions	4 / 50,528 (< 1 %)
Dedicated logic registers	0 / 50,528 (0 %)
Total registers	0
Total pins	13 / 450 (3 %)
Total virtual pins	0
Total memory bits	0 / 594,432 (0 %)
Embedded Multiplier 9-bit elements	0 / 172 (0 %)
Total PLLs	0 / 4 (0 %)



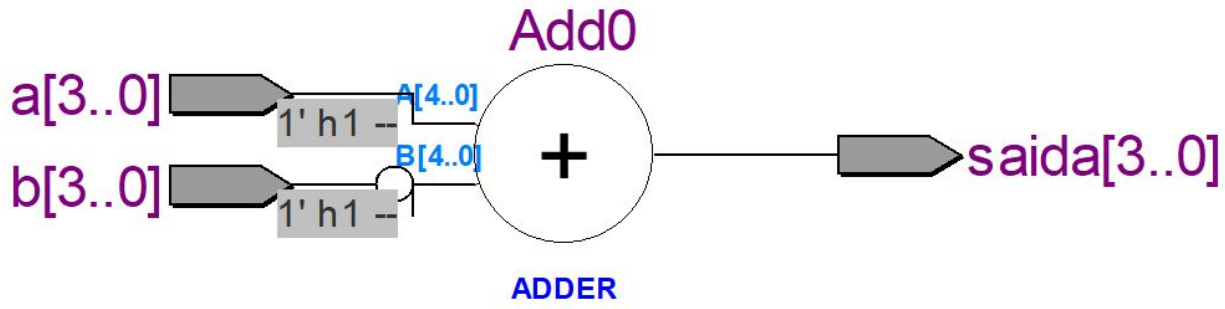
3.1.3 Subtratores de n bits.

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity subtrator_n_bits is
    generic (N: positive := 4);
    port(
        a: in std_logic_vector(N-1 downto 0);
        b: in std_logic_vector(N-1 downto 0);
        saida: out std_logic_vector(N-1 downto 0)
    );
end entity;

architecture arq of subtrator_n_bits is
begin
saida <= std_logic_vector(unsigned(a) - unsigned(b));
end arq;
```

Flow Summary	
Flow Status	Successful - Thu Oct 04 17:52:09 2018
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	UnidadeMDC_Caetano
Top-level Entity Name	subtrator_n_bits
Family	Cyclone II
Device	EP2C50F672C8
Timing Models	Final
Total logic elements	4 / 50,528 (< 1 %)
Total combinational functions	4 / 50,528 (< 1 %)
Dedicated logic registers	0 / 50,528 (0 %)
Total registers	0
Total pins	12 / 450 (3 %)
Total virtual pins	0
Total memory bits	0 / 594,432 (0 %)
Embedded Multiplier 9-bit elements	0 / 172 (0 %)
Total PLLs	0 / 4 (0 %)

Propagation Delay

	Input Port	Output Port	RR	RF	FR	FF
1	b[1]	saida[2]	12.670	12.670	12.670	12.670
2	b[1]	saida[3]	12.415	12.415	12.415	12.415
3	b[2]	saida[3]	12.287	12.287	12.287	12.287
4	a[1]	saida[2]	12.266	12.266	12.266	12.266
5	b[2]	saida[2]	12.143	12.143	12.143	12.143
6	b[1]	saida[1]	12.096	12.096	12.096	12.096
7	a[1]	saida[3]	12.011	12.011	12.011	12.011
8	a[2]	saida[3]	11.973	11.973	11.973	11.973
9	a[2]	saida[2]	11.828	11.828	11.828	11.828
10	a[1]	saida[1]	11.689	11.689	11.689	11.689
11	b[3]	saida[3]	11.329	11.329	11.329	11.329
12	a[3]	saida[3]	10.828	10.828	10.828	10.828
13	b[0]	saida[2]	8.855	8.855	8.855	8.855
14	a[0]	saida[2]	8.808	8.808	8.808	8.808
15	b[0]	saida[1]	8.671	8.671	8.671	8.671
16	a[0]	saida[1]	8.624	8.624	8.624	8.624
17	b[0]	saida[3]	8.600	8.600	8.600	8.600
18	a[0]	saida[3]	8.553	8.553	8.553	8.553
19	b[0]	saida[0]	8.197	8.197	8.197	8.197
20	a[0]	saida[0]	8.149	8.149	8.149	8.149



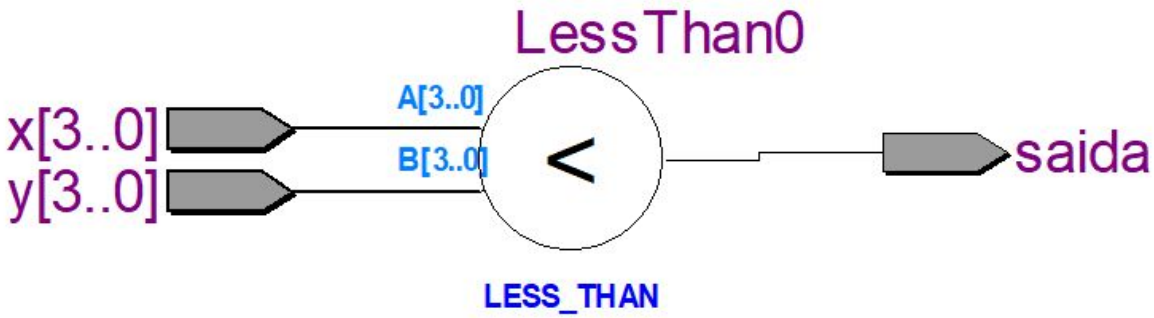
3.1.4 Unidade comparativa X < Y.

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity XmenY is
    generic (n: natural := 4);
    port (
x,y: in std_logic_vector (n-1 downto 0) ;
saida: out std_logic
);
end XmenY;

architecture arq of XmenY is
begin
saida <= '1' when unsigned(x) < unsigned(y) else '0';
end arq;
```

Flow Summary	
Flow Status	Successful - Thu Oct 04 17:55:30 2018
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	UnidadeMDC_Caetano
Top-level Entity Name	XmenY
Family	Cyclone II
Device	EP2C50F672C8
Timing Models	Final
Total logic elements	3 / 50,528 (< 1 %)
Total combinational functions	3 / 50,528 (< 1 %)
Dedicated logic registers	0 / 50,528 (0 %)
Total registers	0
Total pins	9 / 450 (2 %)
Total virtual pins	0
Total memory bits	0 / 594,432 (0 %)
Embedded Multiplier 9-bit elements	0 / 172 (0 %)
Total PLLs	0 / 4 (0 %)

Propagation Delay

	Input Port	Output Port	RR	RF	FR	FF
1	x[0]	saida		14.621	14.621	
2	x[1]	saida		13.225	13.225	
3	x[2]	saida		12.624	12.624	
4	x[3]	saida		6.674	6.674	
5	y[1]	saida	12.894			12.894
6	y[0]	saida	12.459			12.459
7	y[2]	saida	12.068			12.068
8	y[3]	saida	6.840			6.840



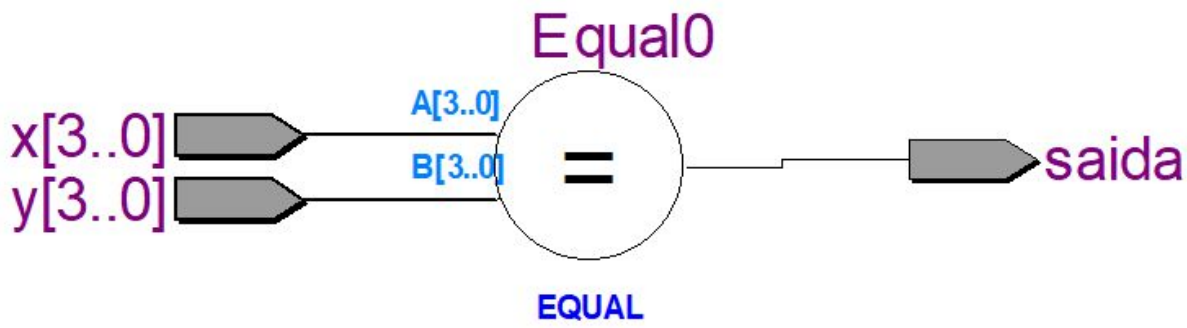
3.1.5 Unidade comparativa X=Y.

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity XigualY is
    generic (n: natural := 4);
    port (
x,y: in std_logic_vector (n-1 downto 0) ;
saida: out std_logic
);
end XigualY;

architecture arq of XigualY is
begin
saida <= '1' when unsigned(x) = unsigned(y) else '0';
end arq;
```

Flow Summary	
Flow Status	Successful - Thu Oct 04 17:57:34 2018
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	UnidadeMDC_Caetano
Top-level Entity Name	XigualY
Family	Cyclone II
Device	EP2C50F672C8
Timing Models	Final
Total logic elements	3 / 50,528 (< 1 %)
Total combinational functions	3 / 50,528 (< 1 %)
Dedicated logic registers	0 / 50,528 (0 %)
Total registers	0
Total pins	9 / 450 (2 %)
Total virtual pins	0
Total memory bits	0 / 594,432 (0 %)
Embedded Multiplier 9-bit elements	0 / 172 (0 %)
Total PLLs	0 / 4 (0 %)

Propagation Delay

	Input Port	Output Port	RR	RF	FR	FF
1	y[2]	saida	12.693	12.693	12.693	12.693
2	x[3]	saida	12.666	12.666	12.666	12.666
3	y[3]	saida	12.623	12.623	12.623	12.623
4	x[2]	saida	12.036	12.036	12.036	12.036
5	y[1]	saida	11.922	11.922	11.922	11.922
6	y[0]	saida	11.660	11.660	11.660	11.660
7	x[0]	saida	7.662	7.662	7.662	7.662
8	x[1]	saida	7.240	7.240	7.240	7.240



3.1.6 Bloco Operativo.

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity datapath is
generic(m: natural := 4);
port (clk : in STD_LOGIC;
      x, y: in STD_LOGIC_VECTOR(m-1 downto 0); --testando com 4 bits
      loadX, loadY, loadS, clearS, clearY, clearX, selX, selY: in STD_LOGIC;
      mdcXY: out STD_LOGIC_VECTOR(m-1 downto 0);
      XmenYflag, XigualYflag : out STD_LOGIC);
end datapath;


architecture arq of datapath is

   --DECLARACAO DE COMPONENTES
   component muxNbits is
   generic (n: natural := 4);
   port (
   s0,s1: in std_logic_vector (n-1 downto 0) ;
   sel : in std_logic;
   saida: out std_logic_vector(n-1 downto 0)
   );
   end component;

   component Register_Nbits is
   generic (N: positive := 4);
   port(
      clk, reset,carga: in std_logic;
      d: in std_logic_vector(N-1 downto 0);
      q: out std_logic_vector(N-1 downto 0)
   );
   end component;

   component subtrator_n_bits is
   generic (N: positive := 1);
   port(
      a: in std_logic_vector(N-1 downto 0);
      b: in std_logic_vector(N-1 downto 0);
      saida: out std_logic_vector(N-1 downto 0)
   );
   end component;
```

```
    component XigualY is
    generic (n: natural := 4);
    port (
    x,y: in std_logic_vector (n-1 downto 0) ;
    saida: out std_logic
    );
    end component;

    component XmenY is
    generic (n: natural := 4);
    port (
    x,y: in std_logic_vector (n-1 downto 0) ;
    saida: out std_logic
    );
    end component;
    --DECLARACAO DE SINAIS
    signal saiMuxX, saiMuxY, saiRegX, saiRegY, saiSub1, saiSub2, saiRegS: std_logic_vector(m-1 downto 0);

begin
    --INSTANCIACAO DOS COMPONENTES
    --'M'(Quantidade de bits) DECLARADO NO GENERIC DO DATAPATH
    muxX: muxNbits generic map(m) port map(x, saiSub2,selX, saiMuxX);
    muxY: muxNbits generic map(m) port map(y, saiSub1,selY, saiMuxY);
    RegY: Register_Nbits generic map(m) port map(clk, clearY, loadY, saiMuxY, saiRegY);
    RegX: Register_Nbits generic map(m) port map(clk, clearX, loadX, saiMuxX, saiRegX);
    Sub1: subtrator_n_bits generic map(m) port map(saiRegY, saiRegX, saiSub1);
    Sub2: subtrator_n_bits generic map(m) port map(saiRegX, saiRegY, saiSub2);
    RegSaida: Register_Nbits generic map(m) port map(clk, clearS, loadS, saiRegX, mdcXY);
    XmenY_PortMap: XmenY generic map(m) port map(saiRegX, saiRegY, XmenYflag);
    XigualY_PortMap: XigualY generic map(m) port map(saiRegX, saiRegY, XigualYflag);
end arq;
```

Clock to Output Times

	Data Port	Clock Port	Rise	Fall	Clock Edge	Clock Reference
1	XigualYflag	clk	9.970	9.970	Rise	clk
2	XmenYflag	clk	9.907	9.907	Rise	clk
3	⌄ mdcXY[*]	clk	7.838	7.838	Rise	clk
1	mdcXY[0]	clk	7.686	7.686	Rise	clk
2	mdcXY[1]	clk	7.664	7.664	Rise	clk
3	mdcXY[2]	clk	7.701	7.701	Rise	clk
4	mdcXY[3]	clk	7.838	7.838	Rise	clk

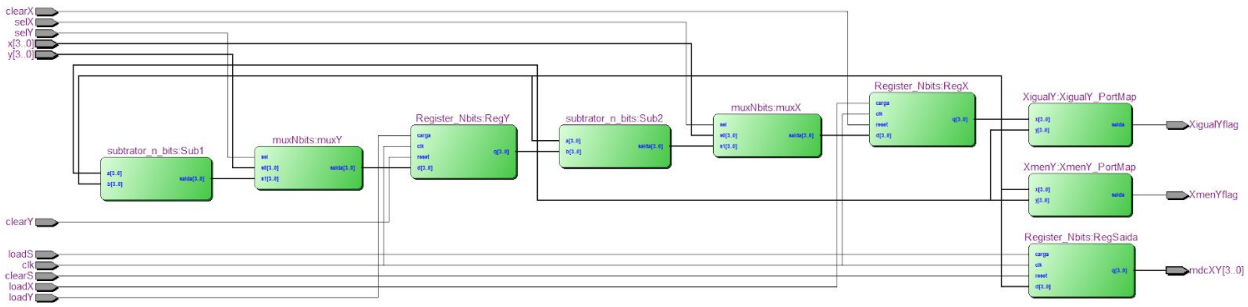
Hold Times

	Data Port	Clock Port	Rise	Fall	Clock Edge	Clock Reference
1	loadS	clk	-0.920	-0.920	Rise	clk
2	loadX	clk	-3.925	-3.925	Rise	clk
3	loadY	clk	-3.986	-3.986	Rise	clk
4	selX	clk	-3.886	-3.886	Rise	clk
5	selY	clk	-3.928	-3.928	Rise	clk
6	⌄ x[*]	clk	-3.554	-3.554	Rise	clk
1	x[0]	clk	-3.643	-3.643	Rise	clk
2	x[1]	clk	-3.644	-3.644	Rise	clk
3	x[2]	clk	-3.554	-3.554	Rise	clk
4	x[3]	clk	-3.936	-3.936	Rise	clk
7	⌄ y[*]	clk	-3.617	-3.617	Rise	clk
1	y[0]	clk	-3.623	-3.623	Rise	clk
2	y[1]	clk	-3.678	-3.678	Rise	clk
3	y[2]	clk	-3.938	-3.938	Rise	clk
4	y[3]	clk	-3.617	-3.617	Rise	clk

Setup Times

	Data Port	Clock Port	Rise	Fall	Clock Edge	Clock Reference
1	loadY	clk	4.252	4.252	Rise	clk
2	⌄ y[*]	clk	4.204	4.204	Rise	clk
1	y[2]	clk	4.204	4.204	Rise	clk
2	y[1]	clk	3.944	3.944	Rise	clk
3	y[0]	clk	3.889	3.889	Rise	clk
4	y[3]	clk	3.883	3.883	Rise	clk
3	⌄ x[*]	clk	4.202	4.202	Rise	clk
1	x[3]	clk	4.202	4.202	Rise	clk
2	x[1]	clk	3.910	3.910	Rise	clk
3	x[0]	clk	3.909	3.909	Rise	clk
4	x[2]	clk	3.820	3.820	Rise	clk
4	selY	clk	4.194	4.194	Rise	clk
5	loadX	clk	4.191	4.191	Rise	clk
6	selX	clk	4.152	4.152	Rise	clk
7	loadS	clk	1.186	1.186	Rise	clk

Flow Summary	
Flow Status	Successful - Thu Oct 04 18:08:29 2018
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	UnidadeMDC_Caetano
Top-level Entity Name	datapath
Family	Cyclone II
Device	EP2C50F672C8
Timing Models	Final
Total logic elements	18 / 50,528 (< 1 %)
Total combinational functions	14 / 50,528 (< 1 %)
Dedicated logic registers	12 / 50,528 (< 1 %)
Total registers	12
Total pins	23 / 450 (5 %)
Total virtual pins	0
Total memory bits	0 / 594,432 (0 %)
Embedded Multiplier 9-bit elements	0 / 172 (0 %)
Total PLLs	0 / 4 (0 %)



3.1.7 Bloco de Controle.

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

entity controle is
port (reset, clk: in STD_LOGIC;
      start, XmenYflag, XigualYflag: in STD_LOGIC;
      loadX, loadY, loadS, clearS, clearX, clearY, selX, selY: OUT STD_LOGIC);
end controle;


architecture arq of controle is
   type InternalState is (S0,S1,S2,S3,S4,S5,S6);
   signal nextState, currentState: InternalState;

begin

   --nextState logic comb
   NSL: process(currentState, XmenYflag, XigualYflag, start) is
   begin
      nextState <= currentState;
      case currentState is
         when S0 =>
            if start = '1' then
               nextState <= S1;
            else
               nextState <= S0;
            end if;
         when S1 =>
            nextState <= S2;
         when S2 =>
            if XigualYflag = '1' then
               nextState <= S6;
            else
               nextState <= S3;
            end if;
         when S3 =>
            if XmenYflag = '1' then
            nextState <= S5;
            else
            nextState <= S4;
            end if;
         when S4 =>
            nextState <= S2;
         when S5 =>
            nextState <= S2;
```

```
            when S6 =>
               nextState <= currentState;
         end case;
      end process;

      --memory element: apenas armazena,define o currentState
      ME: process(clk, reset) is
            begin
               if reset = '1' then
                  currentState <= S0;
               elsif rising_edge(clk) then
                  currentState <= nextState;
               end if;
            end process;

      --output logic comb
      loadX <= '1' when currentState = S4 or currentState = S1 else '0';
      loadY <= '1' when currentState = S1 or currentState = S5 else '0';
      loadS <= '1' when currentState = S6 else '0';
      clearS <= '1' when currentState = S0 else '0';
      clearX <= '1' when currentState = S0 else '0';
      clearY <= '1' when currentState = S0 else '0';
      selX <= '1' when currentState = S4 else '0';
      selY <= '1' when currentState = S5 else '0';

end arq;
```

Flow Summary	
Flow Status	Successful - Thu Oct 04 18:13:04 2018
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	UnidadeMDC_Caetano
Top-level Entity Name	controle
Family	Cyclone II
Device	EP2C50F672C8
Timing Models	Final
Total logic elements	9 / 50,528 (< 1 %)
Total combinational functions	9 / 50,528 (< 1 %)
Dedicated logic registers	7 / 50,528 (< 1 %)
Total registers	7
Total pins	13 / 450 (3 %)
Total virtual pins	0
Total memory bits	0 / 594,432 (0 %)
Embedded Multiplier 9-bit elements	0 / 172 (0 %)
Total PLLs	0 / 4 (0 %)

Setup Times

	Data Port	Clock Port	Rise	Fall	Clock Edge	Clock Reference
1	XigualYflag	clk	4.093	4.093	Rise	clk
2	start	clk	-0.094	-0.094	Rise	clk
3	XmenYflag	clk	-0.334	-0.334	Rise	clk

Hold Times

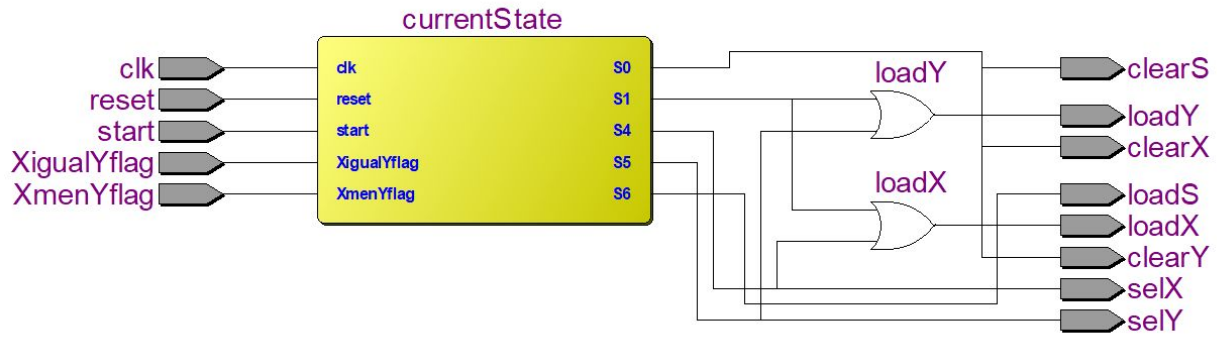
	Data Port	Clock Port	Rise	Fall	Clock Edge	Clock Reference
1	XigualYflag	clk	-3.707	-3.707	Rise	clk
2	XmenYflag	clk	0.602	0.602	Rise	clk
3	start	clk	0.787	0.787	Rise	clk

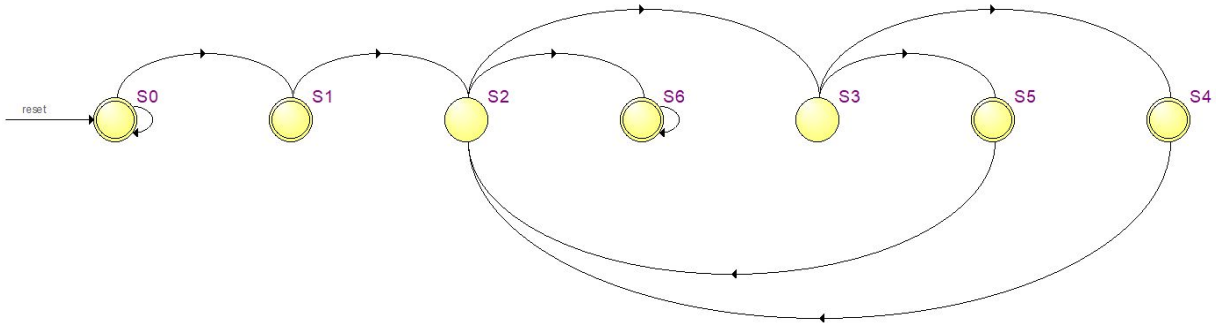
Clock to Output Times

	Data Port	Clock Port	Rise	Fall	Clock Edge	Clock Reference
1	loadX	clk	9.193	9.193	Rise	clk
2	loadY	clk	8.695	8.695	Rise	clk
3	clearX	clk	8.348	8.348	Rise	clk
4	clearS	clk	8.013	8.013	Rise	clk
5	clearY	clk	8.013	8.013	Rise	clk
6	loadS	clk	7.995	7.995	Rise	clk
7	selX	clk	7.663	7.663	Rise	clk
8	selY	clk	7.656	7.656	Rise	clk

Slow Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name	Note
1	694.93 MHz	340.02 MHz	clk	limit due to minimum period restriction (max I/O toggle rate)





3.1.8 O bloco MDC.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

--BLOCO TOPO DO PROJETO
entity mdc is
generic(m: natural := 4);
port(
x,y: in std_logic_vector (m-1 downto 0) ;
mdcxy: out std_logic_vector(m-1 downto 0);
clk,start, reset: in std_logic
);
end entity;



architecture arq of mdc is

--DECLARAÇAO DE COMPONENTES:
   component datapath is
      generic(m: natural := 4);
      port (clk : in STD_LOGIC;
      x, y: in STD_LOGIC_VECTOR(m-1 downto 0); --testando com 4 bits
      loadX, loadY, loadS, clearS, clearY, clearX, selX, selY: in STD_LOGIC;
      mdcXY: out STD_LOGIC_VECTOR(m-1 downto 0);
      XmenYflag, XigualYflag : out STD_LOGIC);
   end component;

   component controle is
      port(reset, clk: IN STD_LOGIC;
      start, XmenYflag, XigualYflag: IN STD_LOGIC;
      loadX, loadY, loadS, clearS, clearX, clearY, selX, selY: OUT STD_LOGIC);
   end component;

--DECLARAÇAO DE SINAIS
signal loadXs, loadYs, loadSs, clearSs, clearYs, clearXs, selXs, selYs: std_logic;
signal XmenYflag, XigualYflag: std_logic;
```

```
--DECLARAÇAO DE SINAIS
signal loadXs, loadYs, loadSs, clearSs, clearYs, clearXs, selXs, selYs: std_logic;
signal XmenYflag, XigualYflag: std_logic;

begin
   --DECLARACAO DATAPATH
   bo: datapath generic map(m) port map(clk, x, y,
   loadXs, loadYs, loadSs,
   clearSs, clearYs, clearXs,
   selXs, selYs,mdcxy,XmenYflag,XigualYflag);
   --DECLARACAO BLOCO DE CONTROLE
   bc: controle port map (reset, clk , start,XmenYflag,XigualYflag,loadXs, loadYs, loadSs, clearSs, clearYs, clearXs, selXs, selYs);
end arq;
```

Flow Summary

Flow Status	Successful - Thu Oct 04 18:18:21 2018
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	UnidadeMDC_Caetano
Top-level Entity Name	mdc
Family	Cyclone II
Device	EP2C50F672C8
Timing Models	Final
Total logic elements	25 / 50,528 (< 1 %)
Total combinational functions	21 / 50,528 (< 1 %)
Dedicated logic registers	19 / 50,528 (< 1 %)
Total registers	19
Total pins	15 / 450 (3 %)
Total virtual pins	0
Total memory bits	0 / 594,432 (0 %)
Embedded Multiplier 9-bit elements	0 / 172 (0 %)
Total PLLs	0 / 4 (0 %)

Slow Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name	Note
1	345.18 MHz	340.02 MHz	clk	limit due to minimum period restriction (max I/O toggle rate)

Setup Times

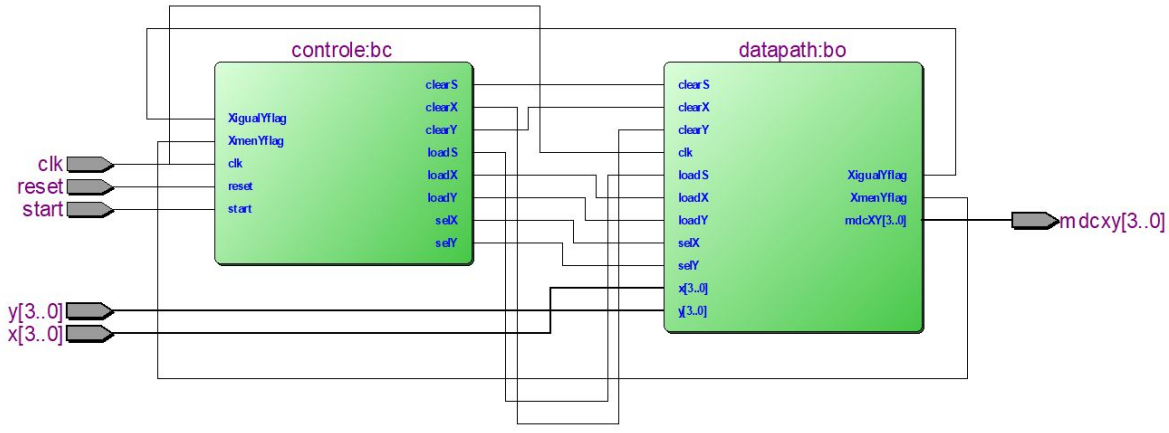
	Data Port	Clock Port	Rise	Fall	Clock Edge	Clock Reference
1	start	clk	0.187	0.187	Rise	clk
2	⌄ y[*]	clk	4.234	4.234	Rise	clk
1	y[0]	clk	3.802	3.802	Rise	clk
2	y[1]	clk	3.854	3.854	Rise	clk
3	y[2]	clk	3.858	3.858	Rise	clk
4	y[3]	clk	4.234	4.234	Rise	clk
3	⌄ x[*]	clk	4.282	4.282	Rise	clk
1	x[0]	clk	0.357	0.357	Rise	clk
2	x[3]	clk	3.808	3.808	Rise	clk
3	x[1]	clk	4.227	4.227	Rise	clk
4	x[2]	clk	4.282	4.282	Rise	clk

Hold Times

	Data Port	Clock Port	Rise	Fall	Clock Edge	Clock Reference
1	start	clk	0.081	0.081	Rise	clk
2	⌄ x[*]	clk	-0.091	-0.091	Rise	clk
1	x[0]	clk	-0.091	-0.091	Rise	clk
2	x[1]	clk	-3.961	-3.961	Rise	clk
3	x[2]	clk	-4.016	-4.016	Rise	clk
4	x[3]	clk	-3.542	-3.542	Rise	clk
3	⌄ y[*]	clk	-3.536	-3.536	Rise	clk
1	y[0]	clk	-3.536	-3.536	Rise	clk
2	y[1]	clk	-3.588	-3.588	Rise	clk
3	y[2]	clk	-3.592	-3.592	Rise	clk
4	y[3]	clk	-3.968	-3.968	Rise	clk

Clock to Output Times

	Data Port	Clock Port	Rise	Fall	Clock Edge	Clock Reference
1	⌄ mdcxy[*]	clk	8.000	8.000	Rise	clk
1	mdcxy[0]	clk	7.641	7.641	Rise	clk
2	mdcxy[1]	clk	8.000	8.000	Rise	clk
3	mdcxy[2]	clk	7.652	7.652	Rise	clk
4	mdcxy[3]	clk	7.983	7.983	Rise	clk



4.Testes de Validação

4.1.1 Registradores de n bits

Arquivo de estímulos usado(register.do):

force /clk 0 0ns, 1 10ns -r 20ns

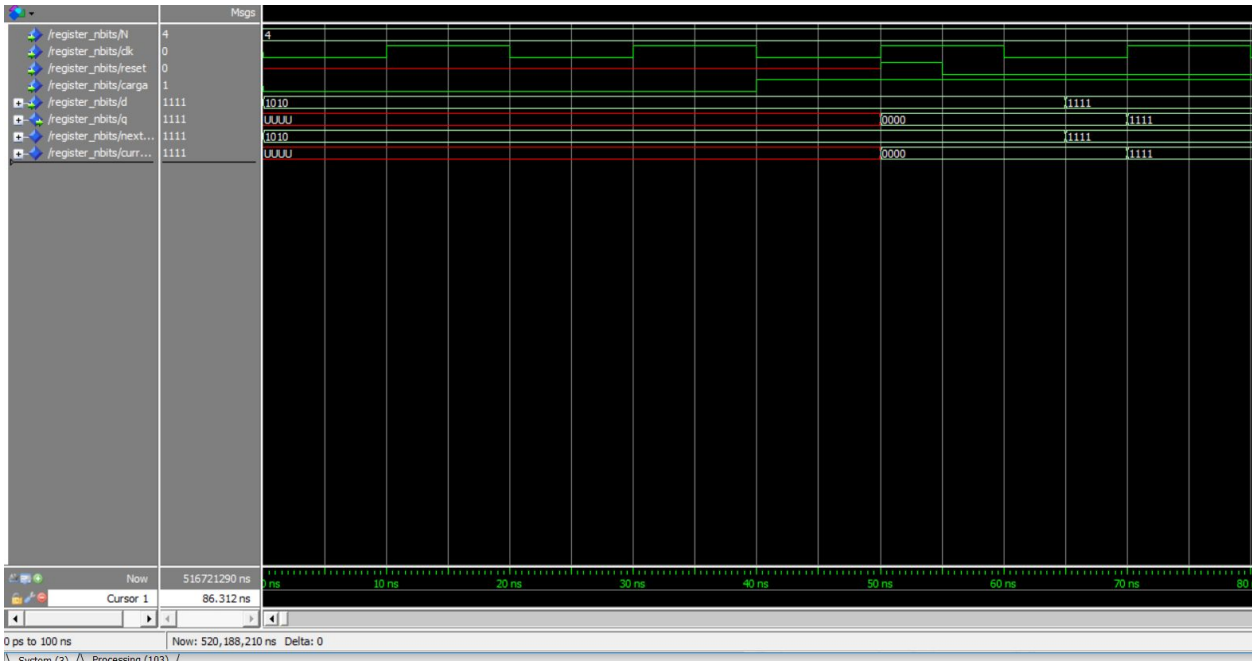
force /d 1010 0ns

force /carga 0 0ns, 1 40ns

force /reset 1 50ns, 0 55ns

force /d 1111 65ns

Resultado obtido:



conforme desejado, o reset assíncrono está funcionando, assim como a entrada de load(o conteúdo é copiado de “d” para “q” na próxima transição positiva de clock quando carga = ‘1’).

4.1.2 Multiplexadores 2:1 de n bits

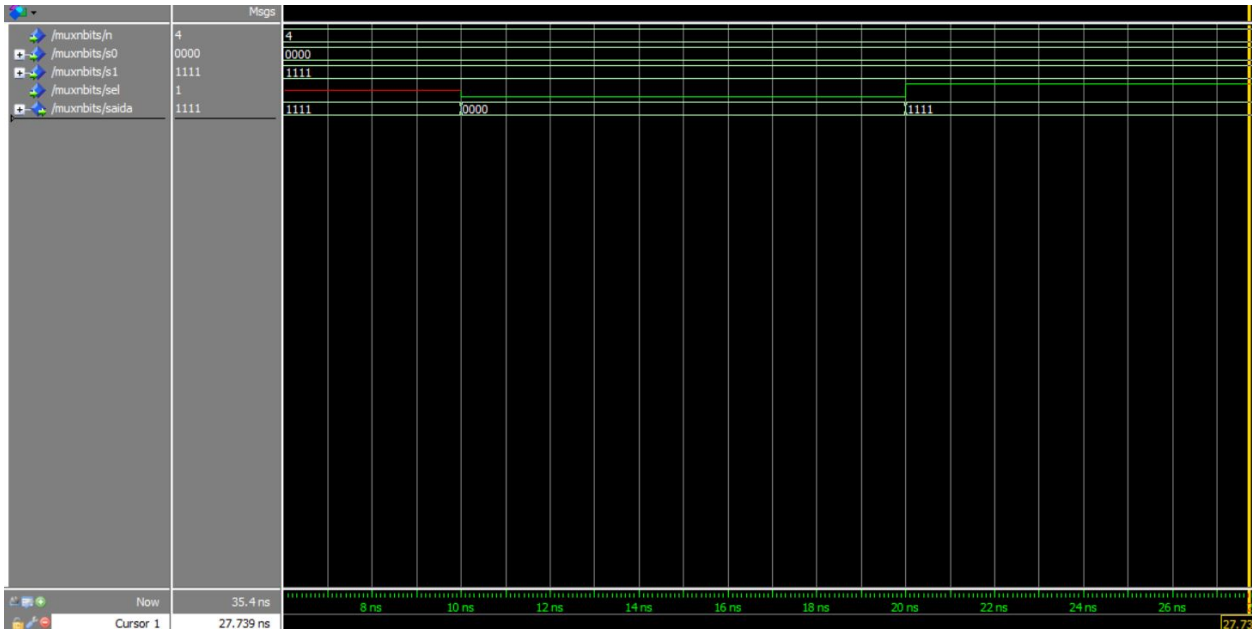
Arquivo de estímulos usado(mux.do):

force /s0 0000 0ns

force /s1 1111 0ns

force /sel 0 10ns, 1 20ns

Resultado obtido:



o multiplexador está funcionando conforme desejado, quando o sinal de seleção é 0 ele copia s0, quando é 1 ele copia s1 para saida.

4.1.3 Subtratores de n bits

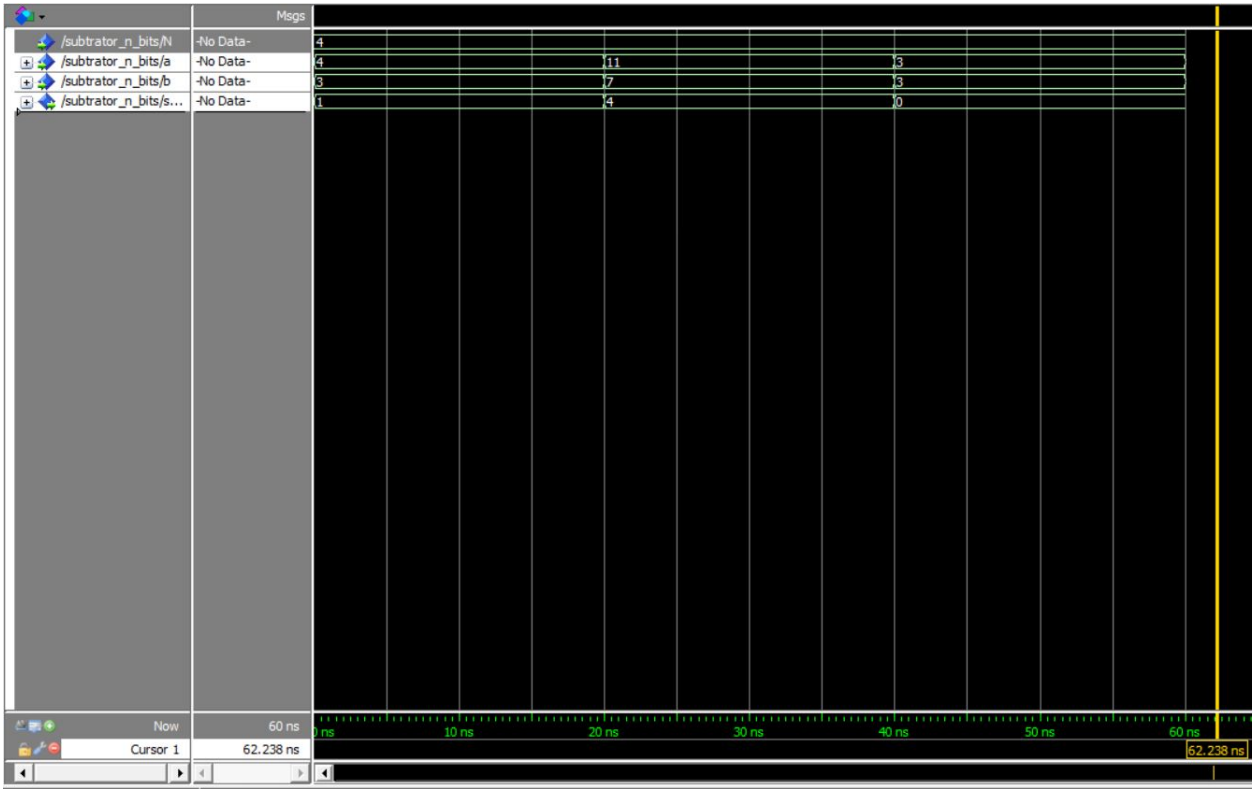
Na simulação o radix foi modificado para unsigned para facilitar a visualização dos dados.

Arquivo de estímulos usado(subtrator.do):

force /a 0100 0ns, 1011 20ns, 0011 40ns, 0000 60ns

force /b 0011 0ns, 0111 20ns, 0011 40ns, 0000 60ns

Resultado obtido:



a - b = saida

4 - 3 = 1

11 - 7 = 4

3 - 3 = 0

4.1.4 Unidade comparativa X<Y

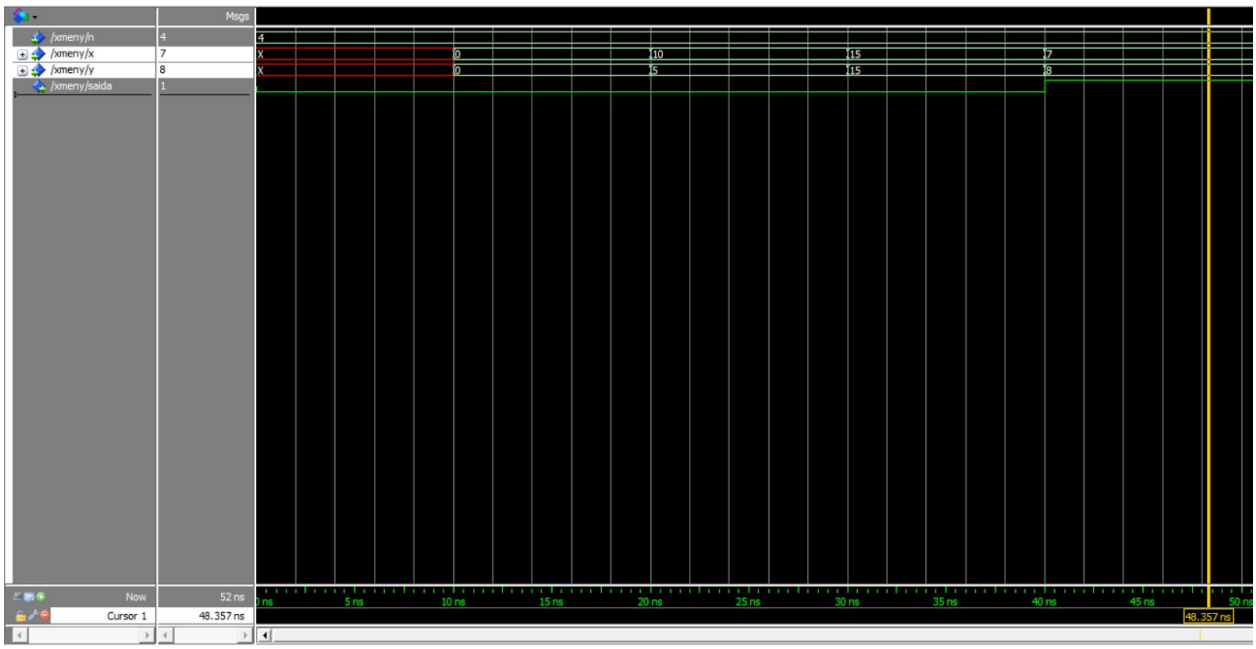
Na simulação o radix foi modificado para unsigned para facilitar a visualização dos dados.

Arquivo de estímulos usado(xmeny.do):

force /x 0000 10ns, 1010 20ns, 1111 30ns, 0111 40ns

force /y 0000 10ns, 0101 20ns, 1111 30ns, 1000 40ns

Resultado obtido:



Entradas(Unsigned)	Saída
x e y	-
0 e 0	0
10 e 5	0
15 e 15	0
7 e 8	1

4.1.5 Unidade comparativa X=Y

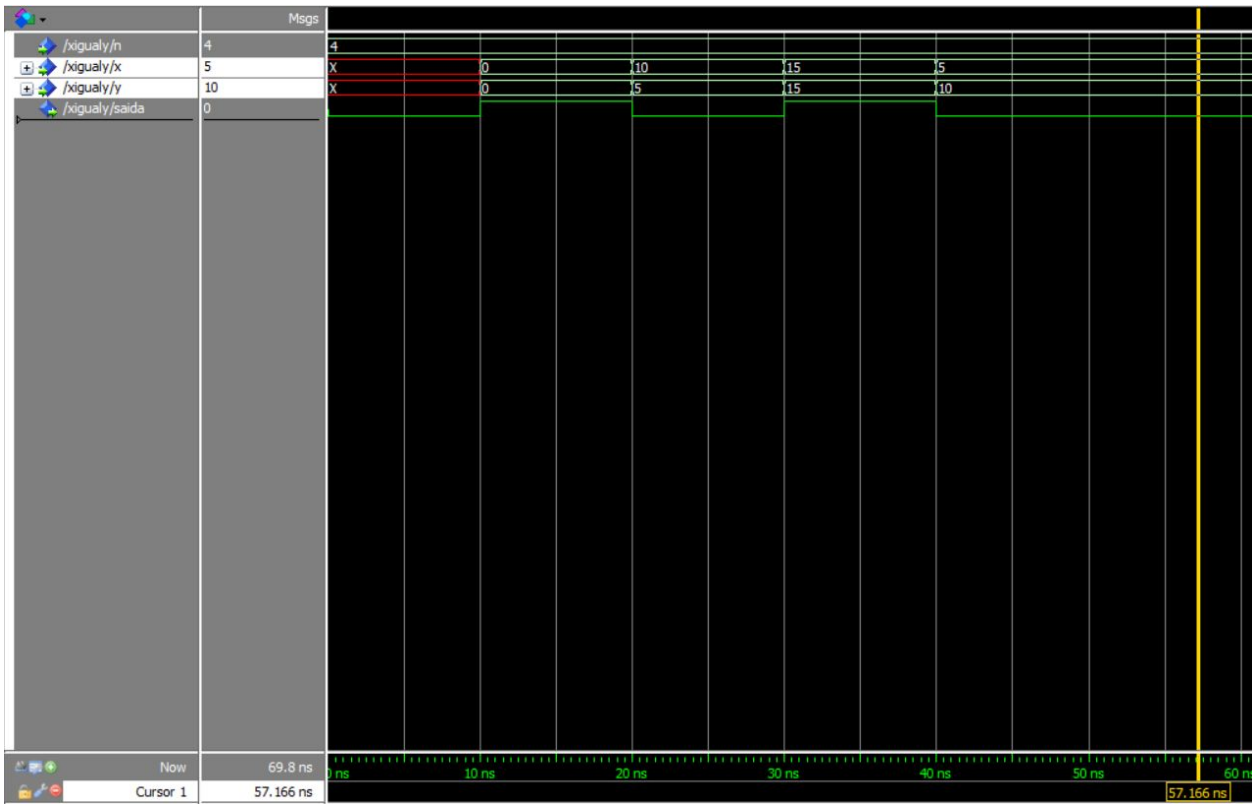
Na simulação o radix foi modificado para unsigned para facilitar a visualização dos dados.

Arquivo de estímulos usado(xigualy.do):

force /x 0000 10ns, 1010 20ns, 1111 30ns, 0101 40ns

force /y 0000 10ns, 0101 20ns, 1111 30ns, 1010 40ns

Resultado obtido:



Entradas(Unsigned)	Saída
x e y	-
0 e 0	1
10 e 5	0
15 e 15	1
5 e 10	0

4.1.6 Bloco Operativo

Na simulação o radix foi modificado para unsigned para facilitar a visualização dos dados.

Arquivo de estímulos usado(datapath.do):

force /clk 0 0ns, 1 10ns -r 20ns

force /x 1000 0ns

force /y 0100 0ns

force /loadX 1 0ns, 0 50ns, 1 100ns, 0 115ns

force /loadY 1 0ns, 0 50ns

force /loadS 1 1000ns

force /clearX 1 0ns, 0 5ns

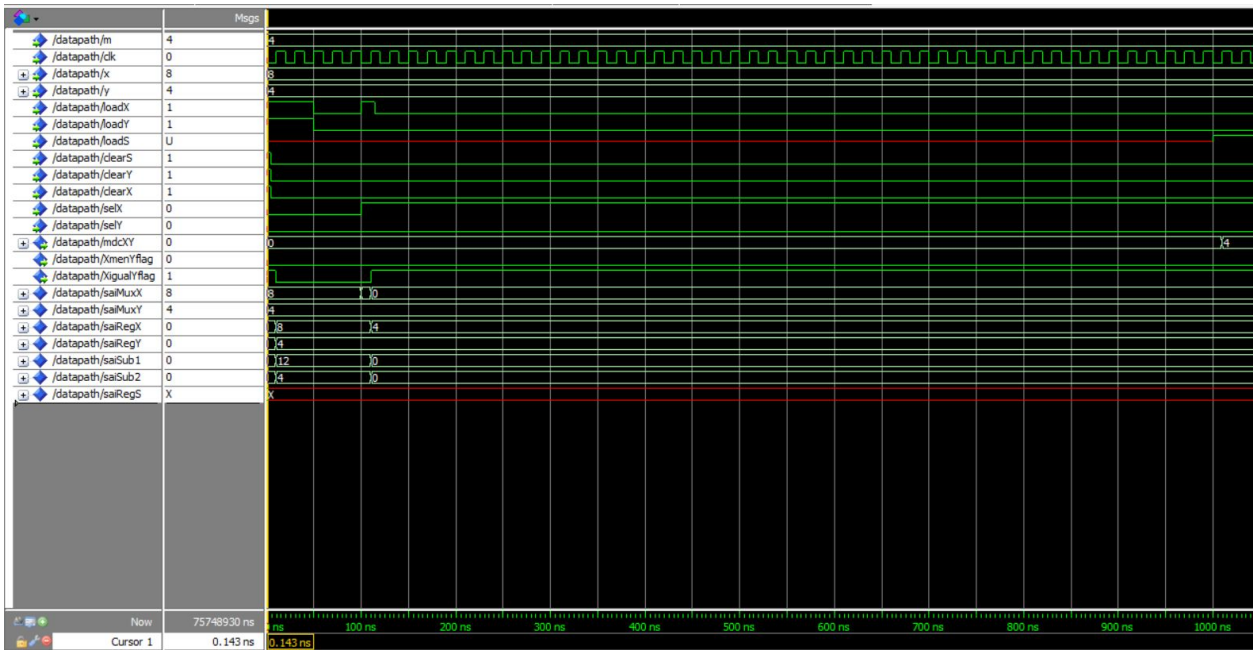
force /clearY 1 0ns, 0 5ns

force /clearS 1 0ns, 0 5ns

force /selX 0 0ns, 1 100ns

force /selY 0 0ns

Resultado obtido:



O datapath está funcionando conforme esperado, entram os números 8 e 4, então XigualYflag = 0, então copia a saída do subtrator em que a saída é 8 - 4, para o registrador x, ou seja selX = 1 e loadX = 1, com isso XigualYflag = 1 e o resultado do mdc é 4.

4.1.7 Bloco de Controle

Na simulação o radix foi modificado para unsigned para facilitar a visualização dos dados.

Arquivo de estímulos usado(controle.do):

force /clk 0 0ns, 1 10ns -r 20ns

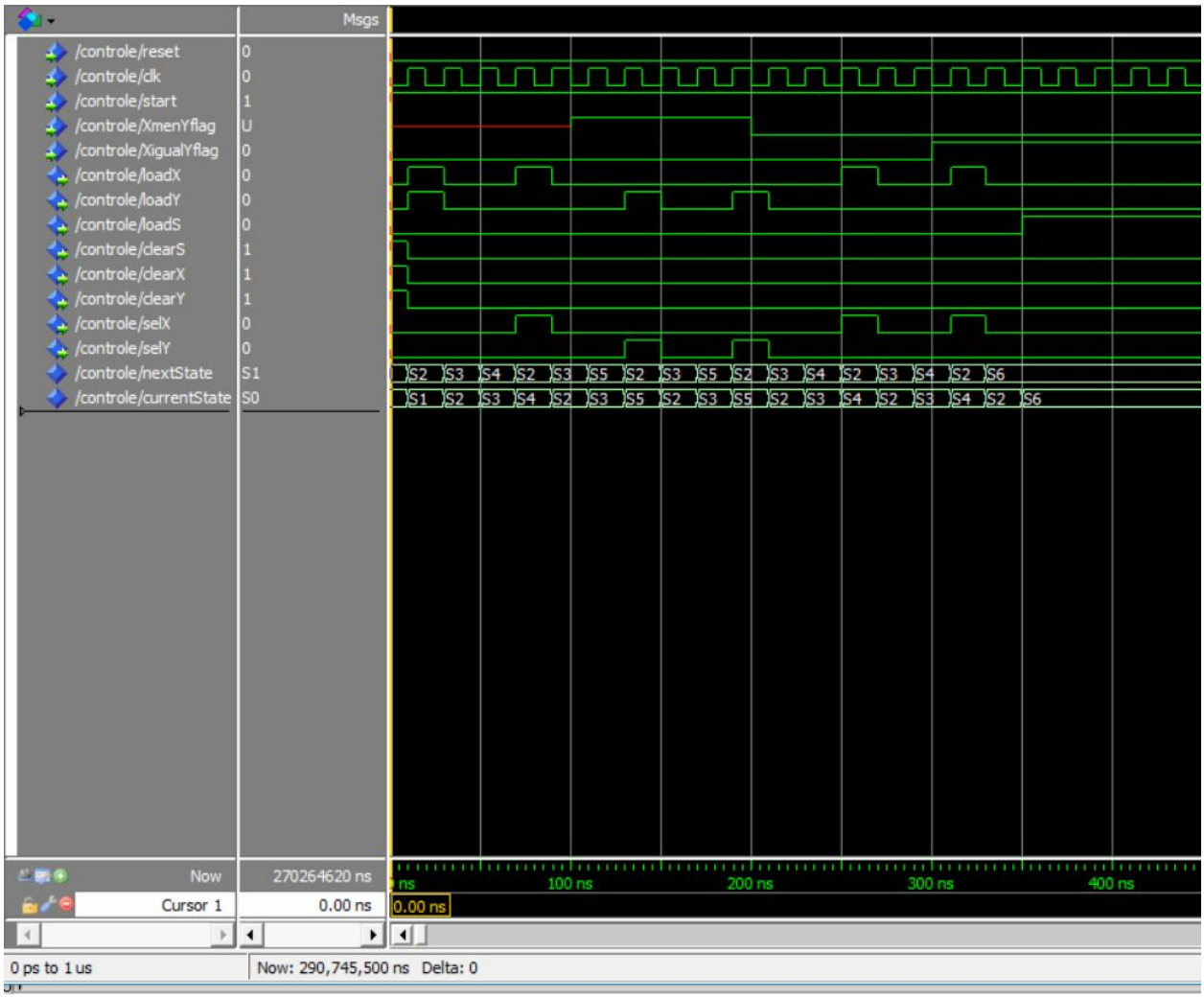
force /reset 0 0ns

force /start 1 0ns

force /XmenYflag 1 100ns, 0 200ns

force /XigualYflag 0 0ns, 1 300ns

Resultado obtido:



A lógica de saída do bloco de controle está funcionando conforme esperado e as transições de nextState também estão funcionando conforme o desejado.

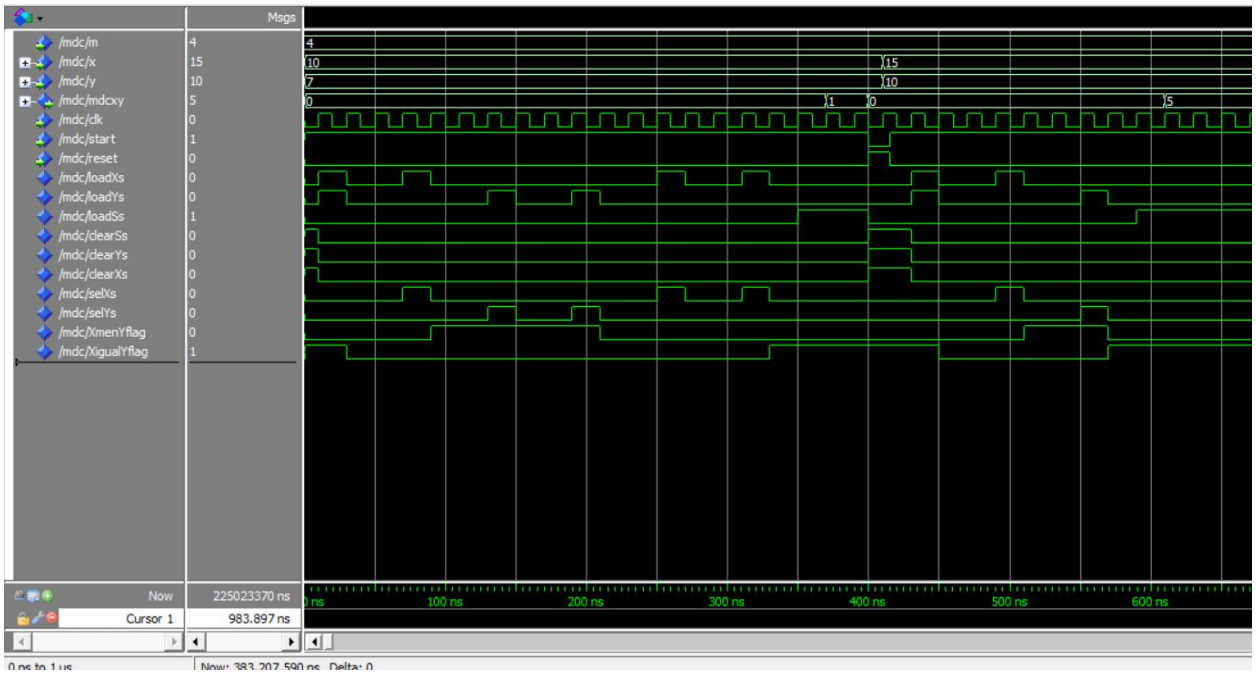
4.1.8 O bloco MDC

Na simulação o radix foi modificado para unsigned para facilitar a visualização dos dados.

Arquivo de estímulos usado(mdc.do):

```
force /clk 0 0ns, 1 10ns -r 20ns

force /x 1010 0ns, 1111 410ns

force /y 0111 0ns, 1010 410ns

force /start 1 0ns, 0 400ns, 1 415ns

force /reset 0 0ns, 1 400ns, 0 415ns
```

Resultado obtido:



apenas um teste com 32 bits mostrando a possibilidade de passar m(número de bits de x e y) como parâmetro:

Arquivo usado(mdc32bits.do):

force /clk 0 0ns, 1 10ns -r 20ns

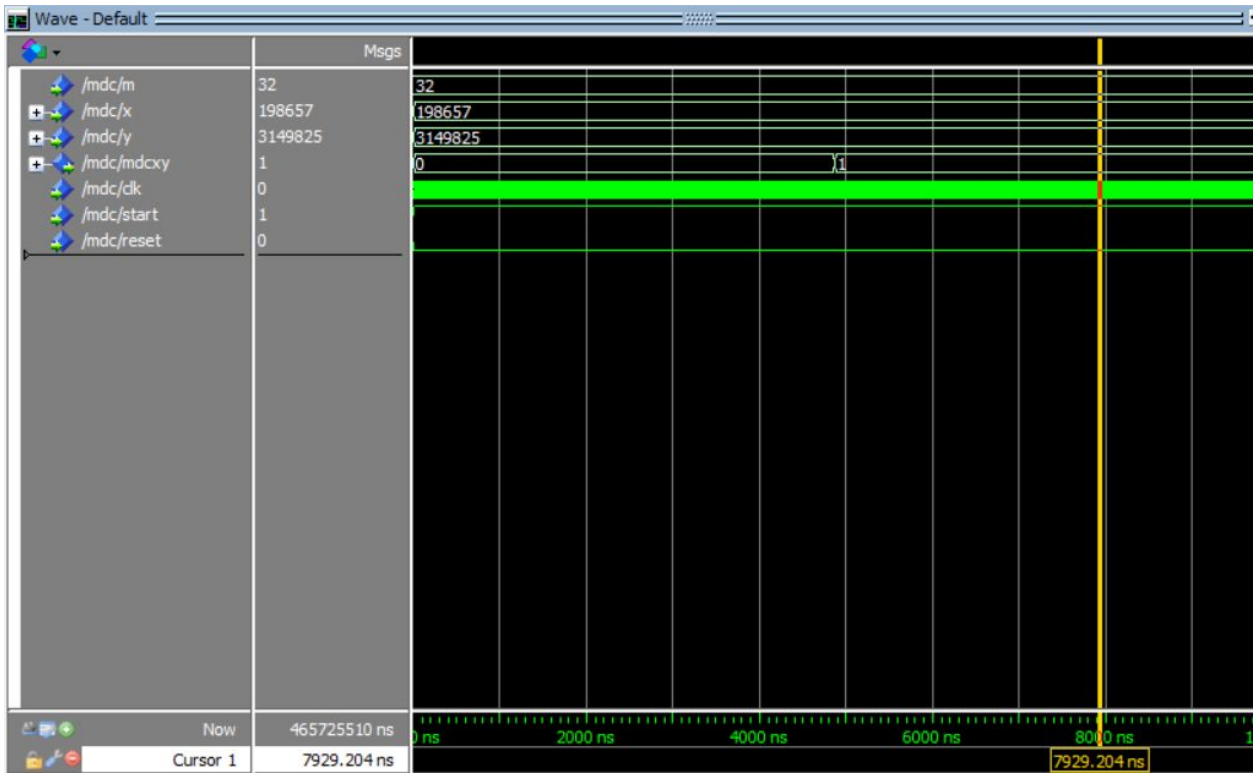
force /x 00000000000000110000100000000001 0ns

force /y 00000000001100000001000000000001 0ns

force /reset 0 0ns

force /start 1 1ns

Resultado obtido:



A unidade MDC está funcionando conforme esperado.