



Técnicas Digitais para Computação - INF01118

Professor: *Fernando R. Nascimento*

Turma: *B*

Aula Prática 09 (AP09)

Nomes:

Felipe de Souza Lahti – matr. 170715

Germano de Mello Andersson – matr. 137719

Porto Alegre, 22 de maio de 2009.

1. Introdução

Em nossa nona aula prática, implementamos um conversor que recebe como entrada sete bits representando um número binário e retorna 8 bits, representando 2 dígitos decimais. O primeiro dígito decimal representa a casa da unidade e o segundo o da dezena, obviamente relativos ao número binário. Para os números com centena, ativamos um 1bit extra. Quando ele está ligado, o número tem centena, quando está desligado não tem. Para o conversor de número binário para dígitos decimais e para o ativador de centena, criamos blocos através de programação VHDL. Este projeto foi desenhado para que possa ser utilizado juntamente ao Kit Altera. Os softwares utilizados para auxílio e implementação foram o Karma e o Max PlusII.

2. Equações Obtidas no Karma e a Tabela Verdade

Para compatibilidade com o Software Karma, o seguinte mapeamento foi feito:

i6= A

i5 = B

i4 = C

i3 = D

i2 = E

i1 = F

Para programar nossos blocos, as seguintes equações foram obtidas do Karma:

$$x2 = !B!CDE!F+!A!BC!D!E!F+!A!BCD!EF+!ABCD!E!F+A!B!CD!F+A!BDE!F+AB!C!DEF+!A!B!C!DF+A!B!D!EF+!BC!DEF+!AB!C!D!F+AB!D!E!F+BC!DE!F+BCDEF+!AB!CDF+ABD!EF$$
$$x3 = !A!BC!E!F+!A!BCD!E+!AB!C!D!EF+!ABC!DEF+!ACD!E!F+A!B!CDE!F+ABC!D!EF+!A!B!C!DE+!AB!CE!F+!A!CDEF+ABCE!F+A!BC!DE+ACDEF+A!B!C!D!E+AB!C!E!F+A!CD!EF$$
$$x4 = !A!B!CD!E!F+!A!BC!D!EF+!A!BCDE!F+!AB!C!DEF+!ABC!D!E!F+!ABCD!EF+A!B!C!DE!F+A!B!CDEF+A!BCD!E!F+AB!C!D!EF+AB!CDE!F+ABC!DEF$$
$$y1 = !B!CDF+!B!CDE+!A!BC!D!E+!BDEF+!ABCD!E+A!B!CEF+A!B!CD+A!BDF+A!BDE+A!CDEF+B!D!EF+!AB!DE+ABC!D+B!C!D!E$$
$$y2 = !A!BCE+!A!BCD+!AB!C!D+A!B!C+ABCD+!ACDE$$
$$y3 = !ABD+!ABC+A!B!C$$
$$y4 = A!BC+AB!C!D!E$$

Para mapearmos estas equações, utilizamos a tabela verdade abaixo:

I6	I5	I4	I3	I2	I1	I0	DEC	Cent	Dez	y4	y3	y2	y1	Unid	x4	x3	x2	x1
0	0	0	0	0	0	0X	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1X	2	0	0	0	0	0	0	2	0	0	1	0
0	0	0	0	1	0	0X	4	0	0	0	0	0	0	4	0	1	0	0
0	0	0	0	1	1	0X	6	0	0	0	0	0	0	6	0	1	1	0
0	0	0	1	0	0	0X	8	0	0	0	0	0	0	8	1	0	0	0
0	0	0	1	0	1	1X	10	0	1	0	0	0	1	0	0	0	0	0
0	0	0	1	1	0	0X	12	0	1	0	0	0	1	2	0	0	1	0
0	0	0	1	1	1	1X	14	0	1	0	0	0	1	4	0	1	0	0
0	0	1	0	0	0	0X	16	0	1	0	0	0	1	6	0	1	1	0
0	0	1	0	0	1	1X	18	0	1	0	0	0	1	8	1	0	0	0
0	0	1	0	1	0	0X	20	0	2	0	0	1	0	0	0	0	0	0
0	0	1	0	1	1	1X	22	0	2	0	0	1	0	2	0	0	1	0
0	0	1	1	0	0	0X	24	0	2	0	0	1	0	4	0	1	0	0
0	0	1	1	0	1	1X	26	0	2	0	0	1	0	6	0	1	1	0
0	0	1	1	1	0	0X	28	0	2	0	0	1	0	8	1	0	0	0
0	0	1	1	1	1	1X	30	0	3	0	0	1	1	0	0	0	0	0
0	1	0	0	0	0	0X	32	0	3	0	0	1	1	2	0	0	1	0
0	1	0	0	0	1	1X	34	0	3	0	0	1	1	4	0	1	0	0
0	1	0	0	1	0	0X	36	0	3	0	0	1	1	6	0	1	1	0
0	1	0	0	1	1	1X	38	0	3	0	0	1	1	8	1	0	0	0
0	1	0	1	0	0	0X	40	0	4	0	1	0	0	0	0	0	0	0
0	1	0	1	0	1	1X	42	0	4	0	1	0	0	2	0	0	1	0
0	1	0	1	1	0	0X	44	0	4	0	1	0	0	4	0	1	0	0
0	1	0	1	1	1	1X	46	0	4	0	1	0	0	6	0	1	1	0
0	1	1	0	0	0	0X	48	0	4	0	1	0	0	8	1	0	0	0
0	1	1	0	0	1	1X	50	0	5	0	1	0	1	0	0	0	0	0
0	1	1	0	1	0	0X	52	0	5	0	1	0	1	2	0	0	1	0
0	1	1	0	1	1	1X	54	0	5	0	1	0	1	4	0	1	0	0
0	1	1	1	0	0	0X	56	0	5	0	1	0	1	6	0	1	1	0
0	1	1	1	0	1	1X	58	0	5	0	1	0	1	8	1	0	0	0
0	1	1	1	1	0	0X	60	0	6	0	1	1	0	0	0	0	0	0
0	1	1	1	1	1	1X	62	0	6	0	1	1	0	2	0	0	1	0
1	0	0	0	0	0	0X	64	0	6	0	1	1	0	4	0	1	0	0
1	0	0	0	0	1	1X	66	0	6	0	1	1	0	6	0	1	1	0
1	0	0	0	1	0	0X	68	0	6	0	1	1	0	8	1	0	0	0
1	0	0	0	1	1	1X	70	0	7	0	1	1	1	0	0	0	0	0
1	0	0	1	0	0	0X	72	0	7	0	1	1	1	2	0	0	1	0
1	0	0	1	0	1	1X	74	0	7	0	1	1	1	4	0	1	0	0
1	0	0	1	1	0	0X	76	0	7	0	1	1	1	6	0	1	1	0
1	0	0	1	1	1	1X	78	0	7	0	1	1	1	8	1	0	0	0
1	0	1	0	0	0	0X	80	0	8	1	0	0	0	0	0	0	0	0
1	0	1	0	0	1	1X	82	0	8	1	0	0	0	2	0	0	1	0
1	0	1	0	1	0	0X	84	0	8	1	0	0	0	4	0	1	0	0
1	0	1	0	1	1	1X	86	0	8	1	0	0	0	6	0	1	1	0
1	0	1	1	0	0	0X	88	0	8	1	0	0	0	8	1	0	0	0
1	0	1	1	0	1	1X	90	0	9	1	0	0	1	0	0	0	0	0
1	0	1	1	1	0	0X	92	0	9	1	0	0	1	2	0	0	1	0
1	0	1	1	1	1	1X	94	0	9	1	0	0	1	4	0	1	0	0
1	1	0	0	0	0	0X	96	0	9	1	0	0	1	6	0	1	1	0
1	1	0	0	0	1	1X	98	0	9	1	0	0	1	8	1	0	0	0
1	1	0	0	1	0	0X	100	1	0	0	0	0	0	0	0	0	0	0
1	1	0	0	1	1	1X	102	1	0	0	0	0	0	2	0	0	1	0
1	1	0	1	0	0	0X	104	1	0	0	0	0	0	4	0	1	0	0
1	1	0	1	0	1	1X	106	1	0	0	0	0	0	6	0	1	1	0
1	1	0	1	1	0	0X	108	1	0	0	0	0	0	8	1	0	0	0
1	1	0	1	1	1	1X	110	1	1	0	0	0	1	0	0	0	0	0
1	1	1	0	0	0	0X	112	1	1	0	0	0	1	2	0	0	1	0
1	1	1	0	0	1	1X	114	1	1	0	0	0	1	4	0	1	0	0
1	1	1	0	1	0	0X	116	1	1	0	0	0	1	6	0	1	1	0
1	1	1	0	1	1	1X	118	1	1	0	0	0	1	8	1	0	0	0
1	1	1	1	0	0	0X	120	1	2	0	0	1	0	0	0	0	0	0
1	1	1	1	0	1	1X	122	1	2	0	0	1	0	2	0	0	1	0
1	1	1	1	1	0	0X	124	1	2	0	0	1	0	4	0	1	0	0
1	1	1	1	1	1	1X	126	1	2	0	0	1	0	6	0	1	1	0

Tabela Verdade de 7 bits

[illegible]
$$|A|BCEIF + |A|BCDIE + |AB|CDEI + |ABCDEF| + |ACDIEF| + |A|BCIDEF + |ABCIDEF| + |A|BICDE + |ABICEF| + |A|CDEF + ABCEIF + A|BCIDE| + ACDEF + A|BICDI|E + ABICI|EF| + A|CDIE|F$$

Karma

Actions Options Help

6

Enter Function

Normal Mode Teaching Mode

Truth Table

A	B	C	D	E	F	S
0	0	0	0	0	0	0
0	0	0	0	0	1	0
0	0	0	0	1	0	0
0	0	0	0	1	1	0
0	0	0	1	0	0	0
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	0
0	0	1	0	1	0	1
0	0	1	0	1	1	1
0	0	1	1	0	0	1
0	0	1	1	0	1	1
0	0	1	1	1	0	1
0	0	1	1	1	1	1
0	1	0	0	0	0	1
0	1	0	0	0	1	1
0	1	0	0	1	0	1
0	1	0	0	1	1	1
0	1	0	1	0	0	0
0	1	0	1	0	1	0
0	1	0	1	1	0	0
0	1	0	1	1	1	0
0	1	1	0	0	0	0
0	1	1	0	0	1	0
0	1	1	0	1	0	0
0	1	1	0	1	1	0
0	1	1	1	0	0	0
0	1	1	1	0	1	0
0	1	1	1	1	0	1
0	1	1	1	1	1	1
1	0	0	0	0	0	1

Karnaugh Map

Column vars A B C

Row vars D E F

	000	001	011	010	110	111	101	100
000	0	0	0	1	0	0	0	1
001	0	0	0	1	0	0	0	1
011	0	1	0	1	0	0	0	1
010	0	1	0	1	0	0	0	1
110	0	1	1	0	0	1	0	1
111	0	1	1	0	0	1	0	1
101	0	1	0	0	0	1	0	1
100	0	1	0	0	0	1	0	1

Primes

- !A!BCE
- !A!BCD
- !A!B!C!D
- A!B!C
- ABCD
- !ACDE

!A!BCE+!A!BCD+!A!B!C!D+A!B!C+ABCD+!ACDE

Karma

Actions Options Help

6

Enter Function

Normal Mode Teaching Mode

Truth Table

A	B	C	D	E	F	S
0	1	1	1	1	0	1
0	1	1	1	1	1	1
1	0	0	0	0	0	1
1	0	0	0	0	1	1
1	0	0	0	1	0	1
1	0	0	0	1	1	1
1	0	0	1	0	0	1
1	0	0	1	0	1	1
1	0	0	1	1	0	1
1	0	0	1	1	1	1
1	0	1	0	0	0	0
1	0	1	0	0	1	0
1	0	1	0	1	0	0
1	0	1	0	1	1	0
1	0	1	1	1	1	0
1	1	0	0	0	0	0
1	1	0	0	0	1	0
1	1	0	0	1	0	0
1	1	0	0	1	1	0
1	1	0	1	0	0	0
1	1	0	1	0	1	0
1	1	0	1	1	0	0
1	1	0	1	1	1	0
1	1	1	0	0	0	0
1	1	1	0	0	1	0
1	1	1	0	1	0	0
1	1	1	0	1	1	0
1	1	1	1	0	0	0
1	1	1	1	0	1	0
1	1	1	1	1	0	0
1	1	1	1	1	1	0

Karnaugh Map

Column vars A B C

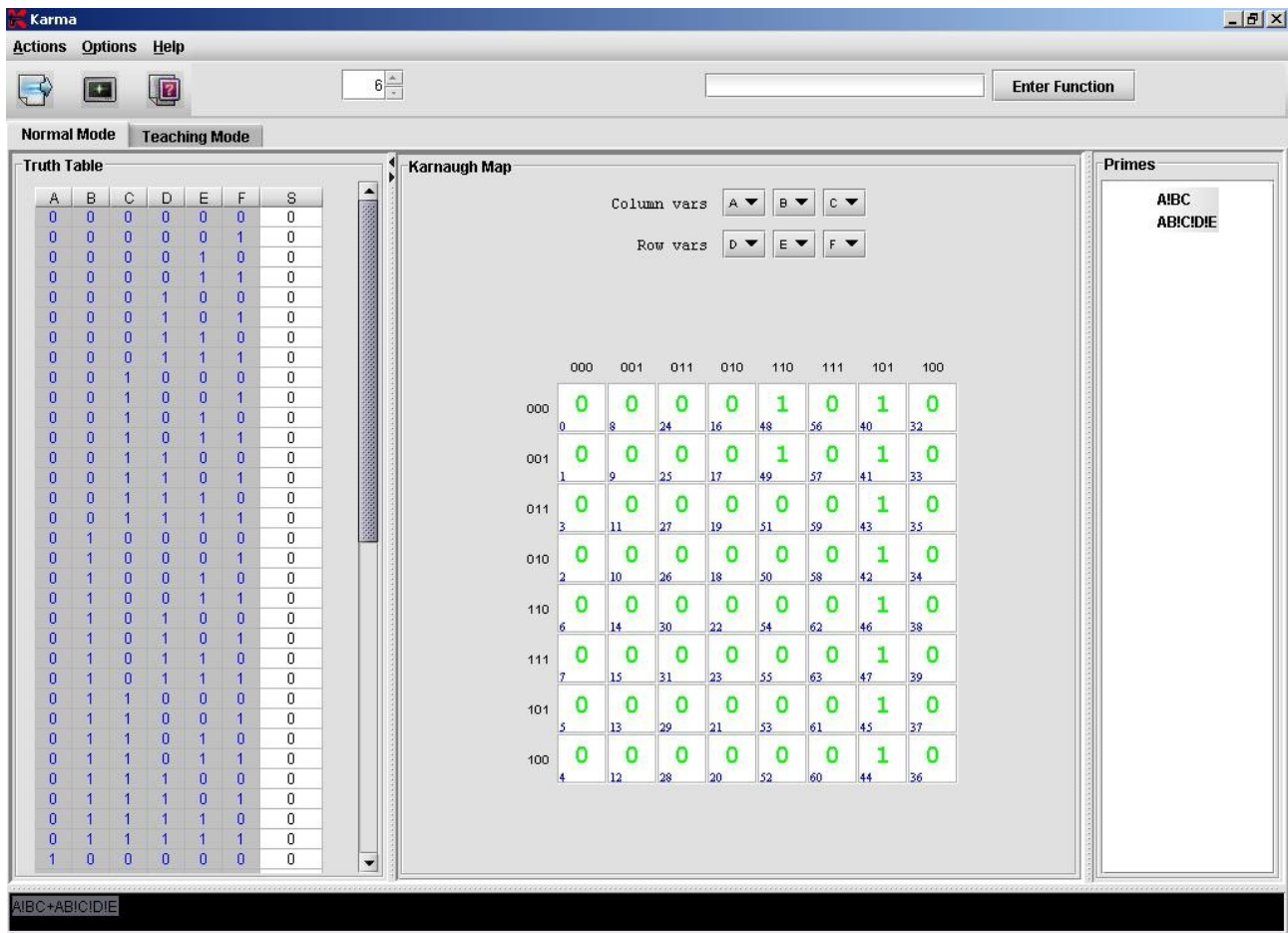
Row vars D E F

	000	001	011	010	110	111	101	100
000	0	0	1	0	0	0	0	1
001	0	0	1	0	0	0	0	1
011	0	0	1	0	0	0	0	1
010	0	0	1	0	0	0	0	1
110	0	0	1	1	0	0	0	1
111	0	0	1	1	0	0	0	1
101	0	0	1	1	0	0	0	1
100	0	0	1	1	0	0	0	1

Primes

- !ABD
- !ABC
- A!B!C

!ABD+!ABC+A!B!C



3. Códigos Fonte em VHDL

Como mencionamos na introdução, foi necessário a criação de dois blocos para implementação do projeto. Segue o código fonte destes dois blocos:

convertabin2dec.vhd:

```
ENTITY conversorBin2Dec IS
    PORT ( I1, I2, I3, I4, I5, I6 : IN      BIT ;
           x2, x3, x4, y1, y2, y3, y4 : OUT    BIT ) ;
END conversorBin2Dec;
ARCHITECTURE LogicFunc OF conversorBin2Dec IS
BEGIN
    x2 <=
        ((NOT I5 AND NOT I4 AND I3 AND I2 AND NOT I1)
        OR
        (NOT I6 AND NOT I5 AND I4 AND NOT I3 AND NOT I2 AND NOT I1)
        OR
```

(NOT I6 AND NOT I5 AND I4 AND I3 AND NOT I2 AND I1)

OR

(NOT I6 AND I5 AND I4 AND I3 AND NOT I2 AND NOT I1)

OR

(I6 AND NOT I5 AND NOT I4 AND I3 AND NOT I1)

OR

(I6 AND NOT I5 AND I3 AND I2 AND NOT I1)

OR

(I6 AND I5 AND NOT I4 AND NOT I3 AND I2 AND I1)

OR

(NOT I6 AND NOT I5 AND NOT I4 AND NOT I3 AND I1)

OR

(I6 AND NOT I5 AND NOT I3 AND NOT I2 AND I1)

OR

(NOT I5 AND I4 AND NOT I3 AND I2 AND I1)

OR

(NOT I6 AND I5 AND NOT I4 AND NOT I3 AND NOT I1)

OR

(I6 AND I5 AND NOT I3 AND NOT I2 AND NOT I1)

OR

(I5 AND I4 AND NOT I3 AND I2 AND NOT I1)

OR

(I5 AND I4 AND I3 AND I2 AND I1)

OR

(NOT I6 AND I5 AND NOT I4 AND I3 AND I1)

OR

(I6 AND I5 AND I3 AND NOT I2 AND I1));

x3 <=

((NOT I6) AND (NOT I5) AND I4 AND (NOT I2) AND (NOT I1)) OR

((NOT I6) AND (NOT I5) AND I4 AND I3 AND (NOT I2)) OR

((NOT I6) AND I5 AND (NOT I4) AND (NOT I3) AND (NOT I2) AND I1) OR

((NOT I6) AND I5 AND I4 AND (NOT I3) AND I2 AND I1) OR

((NOT I6) AND I4 AND I3 AND (NOT I2) AND (NOT I1)) OR

(I6 AND (NOT I5) AND (NOT I4) AND I3 AND I2 AND (NOT I1)) OR

(I6 AND I5 AND I4 AND (NOT I3) AND (NOT I2) AND I1) OR

((NOT I6) AND (NOT I5) AND (NOT I4) AND (NOT I3) AND I2) OR

((NOT I6) AND I5 AND (NOT I4) AND I2 AND (NOT I1)) OR

((NOT I6) AND (NOT I4) AND I3 AND I2 AND I1) OR

(I6 AND I5 AND I4 AND I2 AND (NOT I1)) OR

(I6 AND (NOT I5) AND I4 AND (NOT I3) AND I2) OR

(I6 AND I4 AND I3 AND I2 AND I1) OR

(I6 AND (NOT I5) AND (NOT I4) AND (NOT I3) AND (NOT I2)) OR

(I6 AND I5 AND (NOT I4) AND (NOT I2) AND (NOT I1)) OR

(I6 AND (NOT I4) AND I3 AND (NOT I2) AND I1));

x4 <=


```

(((NOT I6) AND (NOT I5) AND (NOT I4) AND I3 AND (NOT I2) AND (NOT I1) ) OR
((NOT I6) AND (NOT I5) AND I4 AND (NOT I3) AND (NOT I2) AND I1) OR
((NOT I6) AND (NOT I5) AND I4 AND I3 AND I2 AND (NOT I1) ) OR
((NOT I6) AND I5 AND (NOT I4) AND (NOT I3) AND I2 AND I1) OR
((NOT I6) AND I5 AND I4 AND (NOT I3) AND (NOT I2) AND (NOT I1) ) OR
((NOT I6) AND I5 AND I4 AND I3 AND (NOT I2) AND I1) OR
(I6 AND (NOT I5) AND (NOT I4) AND (NOT I3) AND I2 AND (NOT I1) ) OR
(I6 AND (NOT I5) AND (NOT I4) AND I3 AND I2 AND I1) OR
(I6 AND (NOT I5) AND I4 AND I3 AND (NOT I2) AND (NOT I1) ) OR
(I6 AND I5 AND (NOT I4) AND (NOT I3) AND (NOT I2) AND I1) OR
(I6 AND I5 AND (NOT I4) AND I3 AND I2 AND (NOT I1) ) OR
(I6 AND I5 AND I4 AND (NOT I3) AND I2 AND I1));

```

y1 <=

```

(((NOT I5) AND (NOT I4) AND I3 AND I1) OR
((NOT I5) AND (NOT I4) AND I3 AND I2) OR
((NOT I6) AND (NOT I5) AND I4 AND (NOT I3) AND (NOT I2) ) OR
((NOT I5) AND I3 AND I2 AND I1) OR
((NOT I6) AND I5 AND I4 AND I3 AND (NOT I2) ) OR
(I6 AND (NOT I5) AND (NOT I4) AND I2 AND I1) OR
(I6 AND (NOT I5) AND (NOT I4) AND I3) OR
(I6 AND (NOT I5) AND I3 AND I1) OR
(I6 AND (NOT I5) AND I3 AND I2) OR
(I6 AND (NOT I4) AND I3 AND I2 AND I1) OR
(I5 AND (NOT I3) AND (NOT I2) AND I1) OR
((NOT I6) AND I5 AND (NOT I3) AND I2) OR
(I6 AND I5 AND I4 AND (NOT I3) ) OR
(I5 AND (NOT I4) AND (NOT I3) AND (NOT I2)));

```

y2 <=

```

(((NOT I6) AND (NOT I5) AND I4 AND I2) OR
((NOT I6) AND (NOT I5) AND I4 AND I3) OR
((NOT I6) AND I5 AND (NOT I4) AND (NOT I3) ) OR
(I6 AND (NOT I5) AND (NOT I4) ) OR
(I6 AND I5 AND I4 AND I3) OR
((NOT I6) AND I4 AND I3 AND I2));

```

y3 <=

```

(((NOT I6) AND I5 AND I3) OR
((NOT I6) AND I5 AND I4) OR
(I6 AND (NOT I5) AND (NOT I4)));

```

y4 <=

```

((I6 AND (NOT I5) AND I4) OR
(I6 AND I5 AND (NOT I4) AND (NOT I3) AND (NOT I2)));

```

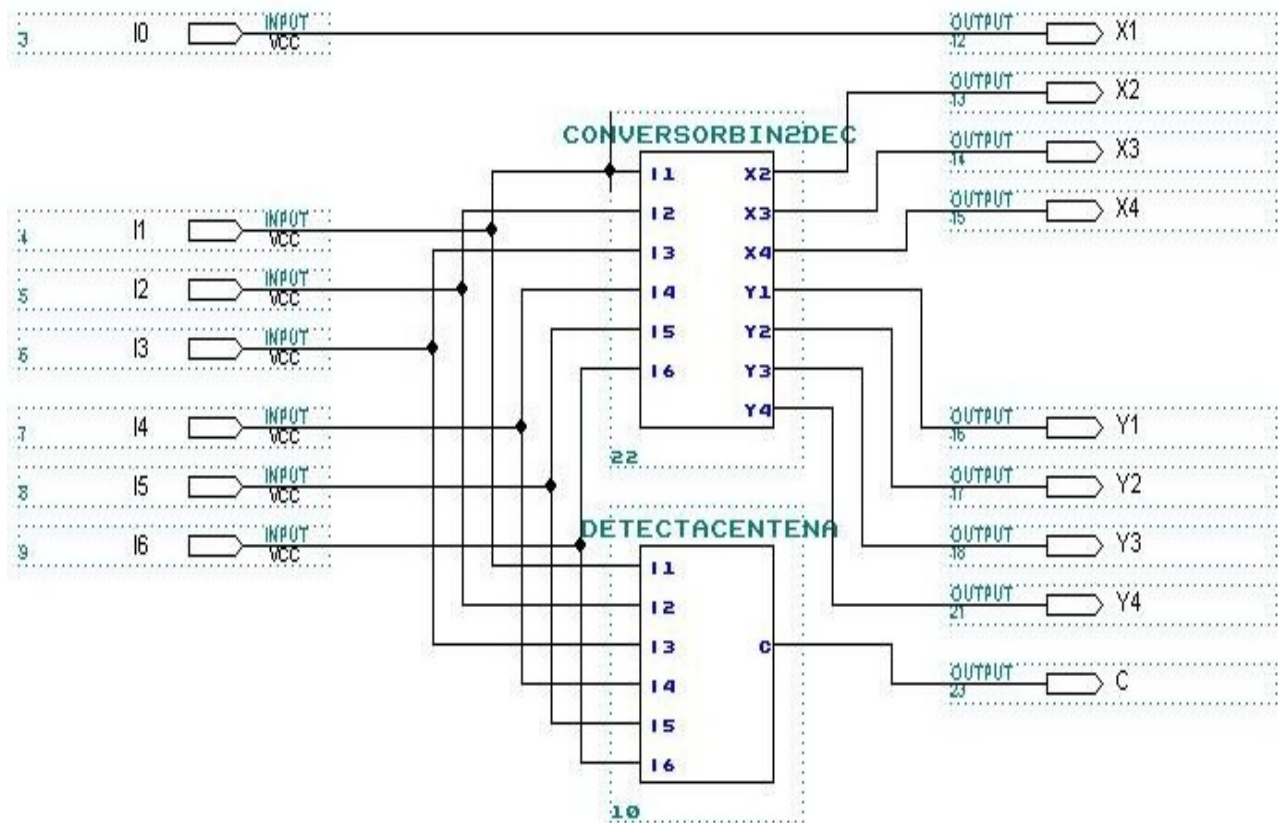
END LogicFunc ;

detectacentena.vhd:

```
ENTITY detectacentena IS
    PORT ( I1, I2, I3, I4, I5, I6 : IN      BIT ;
           c : OUT  BIT ) ;
END detectacentena;
ARCHITECTURE LogicFunc OF detectacentena IS
BEGIN
    c <=
        (I6 AND I5 AND (NOT I4) AND (NOT I3) AND I2 AND (NOT I1)) OR
        (I6 AND I5 AND (NOT I4) AND (NOT I3) AND I2 AND I1) OR
        (I6 AND I5 AND (NOT I4) AND I3 AND (NOT I2) AND (NOT I1)) OR
        (I6 AND I5 AND (NOT I4) AND I3 AND (NOT I2) AND I1) OR
        (I6 AND I5 AND (NOT I4) AND I3 AND I2 AND (NOT I1)) OR
        (I6 AND I5 AND (NOT I4) AND I3 AND I2 AND I1) OR
        (I6 AND I5 AND I4 AND (NOT I3) AND (NOT I2) AND (NOT I1)) OR
        (I6 AND I5 AND I4 AND (NOT I3) AND (NOT I2) AND I1) OR
        (I6 AND I5 AND I4 AND (NOT I3) AND I2 AND (NOT I1)) OR
        (I6 AND I5 AND I4 AND (NOT I3) AND I2 AND I1) OR
        (I6 AND I5 AND I4 AND I3 AND (NOT I2) AND (NOT I1)) OR
        (I6 AND I5 AND I4 AND I3 AND (NOT I2) AND I1) OR
        (I6 AND I5 AND I4 AND I3 AND I2 AND (NOT I1)) OR
        (I6 AND I5 AND I4 AND I3 AND I2 AND I1);
END LogicFunc ;
```

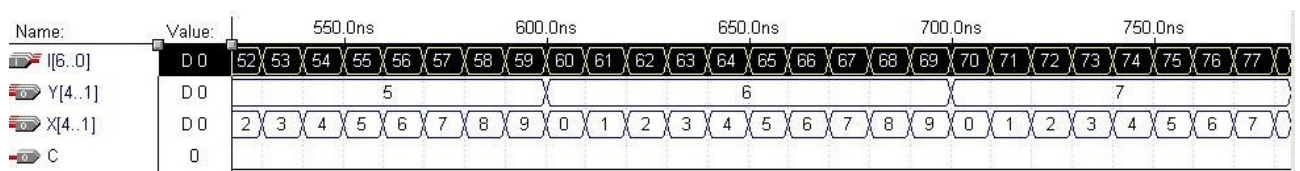
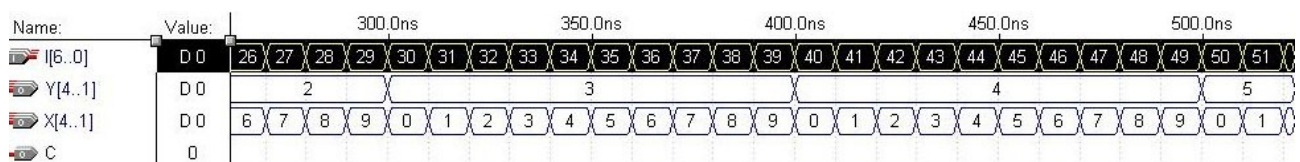
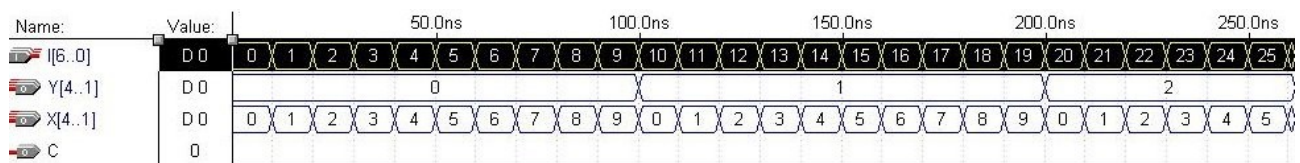
4. Blocos Gerados e o Circuito Completo

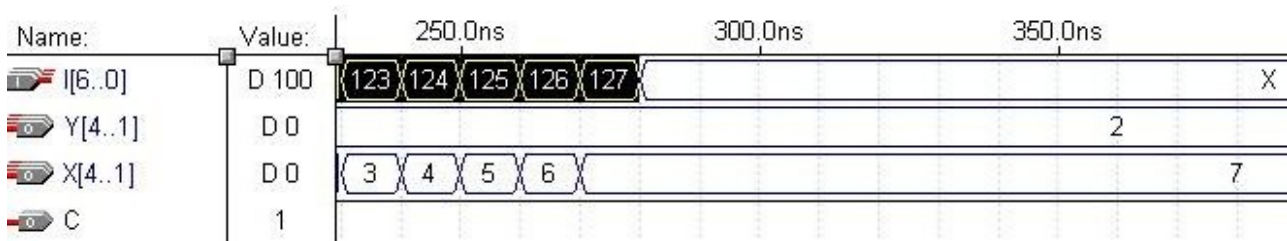
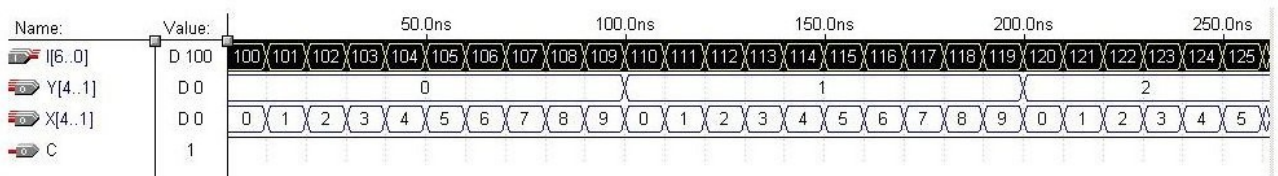
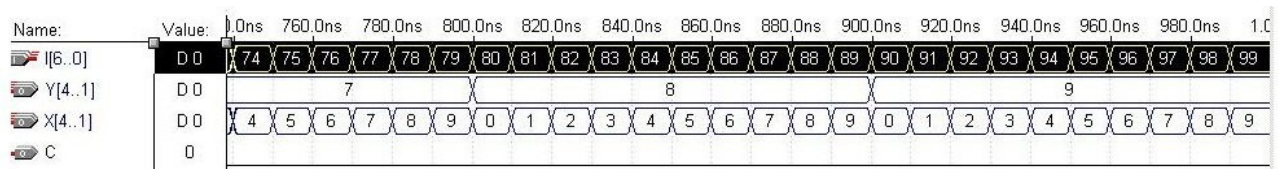
Ao compilarmos estes códigos VHDL, obtivemos os blocos necessários para montarmos nosso conversor binário para decimal. Abaixo segue a imagem do circuito gerado, com as adições necessárias (x1 recebe I0)



5. Testes exaustivos

Para ter a certeza que nosso projeto havia sido implementado corretamente, testamos todos valores possíveis (0 – 127). As imagens abaixo exibem o resultado obtido corretamente:





4. Conclusão

Este foi mais um laboratório interessante. A utilização do software Max Plus e a programação em VHDL começa a nos motivar mais sobre os assuntos aprendidos em aula. Projetar este conversor nos mostrou o quanto estamos absorvendo a matéria, pois não encontramos dificuldade nos detalhes da implementação. Encontramos dificuldade para realizar todas tarefas sugeridas no relatório no período em que estávamos no laboratório o que não permitiu que mostrássemos ao professor a utilização do projeto no Kit Altera.