



Sistemas Lógicos (MiEI) – 2013/2014 – 1º Semestre

Relatório Trabalho Final

Sistema de Alarme

$$2 * A + 1 * B \neq 4.$$

Realizado por:

Luis Silva, nº 34535

Ricardo Cruz, nº 34951

Ricardo Silva, nº 41951

Índice

| | | |
|-----|--|----|
| 1 | Introdução..... | 3 |
| 2 | Análise..... | 4 |
| 3 | Síntese – Parte de Dados..... | 5 |
| 3.1 | Módulo MUX | 6 |
| 3.2 | Módulo Somador..... | 6 |
| 3.3 | Módulo REG_C..... | 7 |
| 3.4 | Módulo OPER K_3 | 8 |
| 3.5 | Módulo F | 8 |
| 4 | Síntese – Parte de Controlo..... | 9 |
| 4.1 | Diagrama de Estados..... | 9 |
| 4.2 | Tabela de Transição de Estados..... | 9 |
| 4.3 | Codificação de Estados..... | 10 |
| 4.4 | Tabela de Transição de Estados codificados, saídas e entradas dos Flip-flops. | 11 |
| 4.5 | Expressões simplificadas das saídas por Mapas de Karnaugh | 12 |
| 4.6 | Expressões simplificadas das entradas dos Flip-flops por Mapas de Karnaugh | 14 |
| 4.7 | Esquemático da Parte de Controlo | 17 |
| 5 | Implementação na FPGA Spartan 3E..... | 18 |
| 6 | Validação através de simulações | 19 |
| 6.1 | Definição das situações para teste | 19 |
| 6.2 | Resultados das simulações..... | 19 |
| 7 | Resultados experimentais..... | 20 |
| 8 | Conclusões e Observações | 21 |

1 Introdução

Este trabalho tem por objectivo, a implementação de um sistema de alarme, que é activado de acordo com uma função previamente definida. Para tal foi necessário a definição de três parâmetros (K_1 , K_2 , K_3) e um operador ($Oper$). Estes parâmetros foram definidos utilizando uma função definida pelos docentes.

Sendo que o resto da divisão da soma dos números de aluno, por 7, define os K_1 , K_2 , K_3 e o resto da divisão da soma dos números de aluno, por 6 define o $Oper$, ou seja o operador a utilizar. Assim sendo foram obtidos os seguintes valores: $K_1 = 2$, $K_2 = 1$ e $K_3 = 4$, e como operador, $oper = ' \neq '$. Ficando a equação completa a implementar: $2 * A + 1 * B \neq 4$.

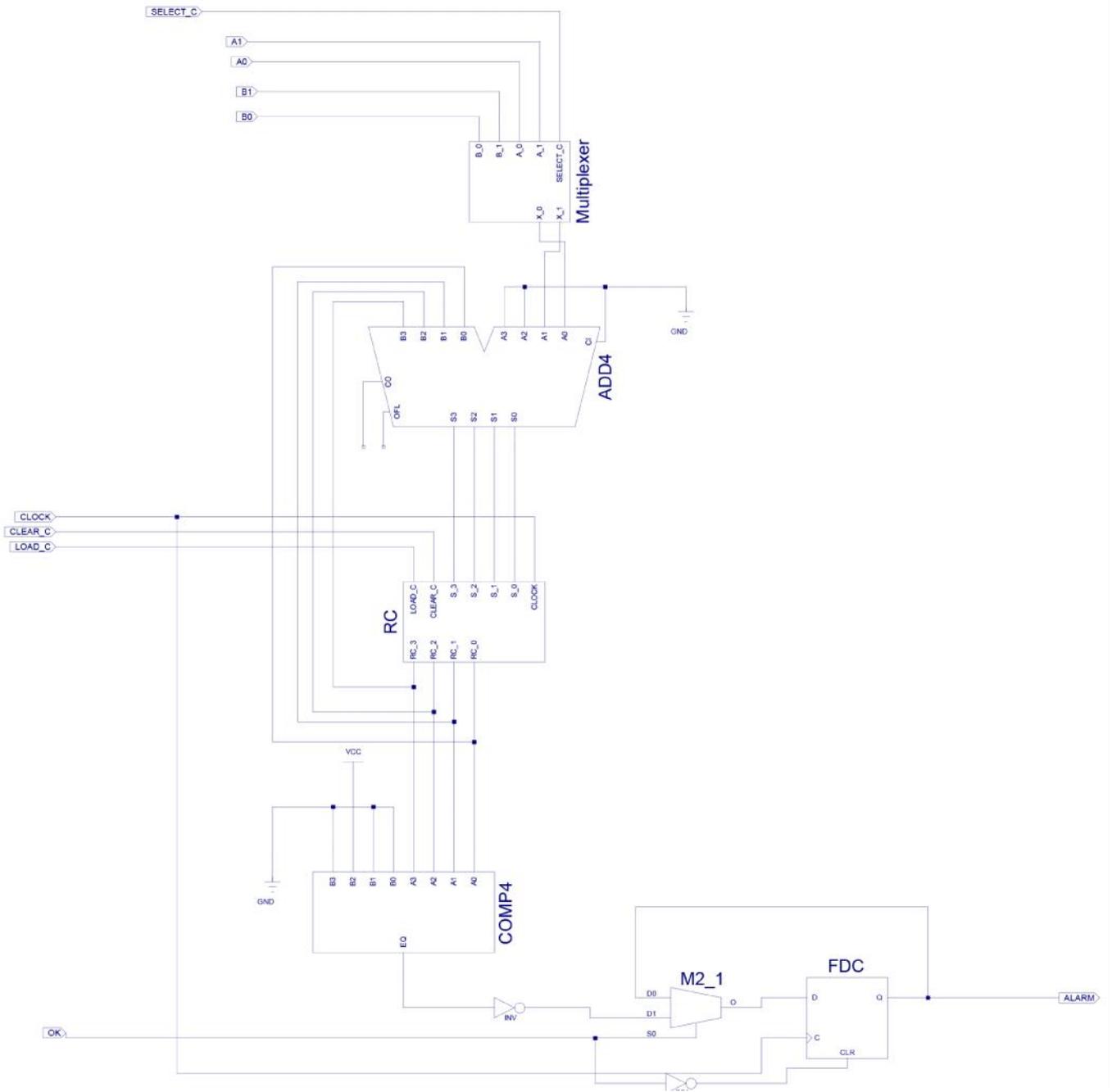
Para realizar este sistema de alarme, existe a necessidade de criar uma parte de controlo, que é responsável pelo comportamento do sistema, e uma parte de dados que implementa o real funcionamento do alarme.

2 Análise

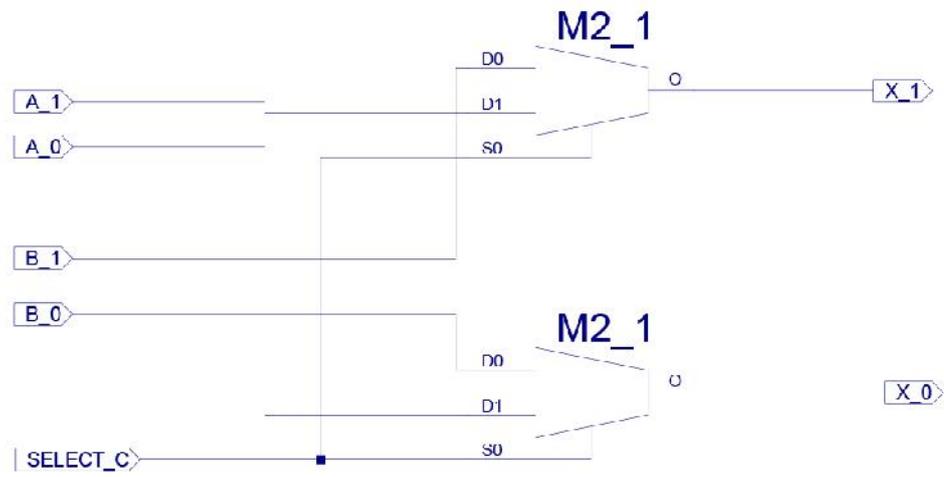
Na elaboração deste projecto, foi seguida a arquitectura recomendada no enunciado, adotando um método de continuidade com as aulas praticas.

Assim o sistema foi dividido em duas partes mais gerais, o *controlo* e os *dados*. Ficando o controlo encarregue de gerir os estados e as transições entre os mesmos, e informando a parte de dados acerca das operações a efectuar. Quanto à parte de dados temos que esta trata de todas as operações e cálculos necessários para apresentar o resultado. Para esta parte foram utilizados os módulos RC e RCI que foram desenvolvidos no decorrer das aulas praticas como módulos de registo, bem como somador, comparador, multiplexers e flip flops da biblioteca do xlinx.

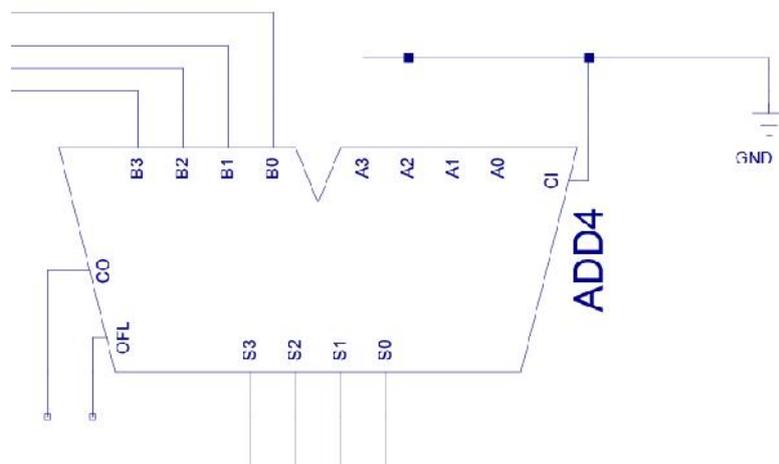
3 Síntese – Parte de Dados



3.1 Módulo MUX

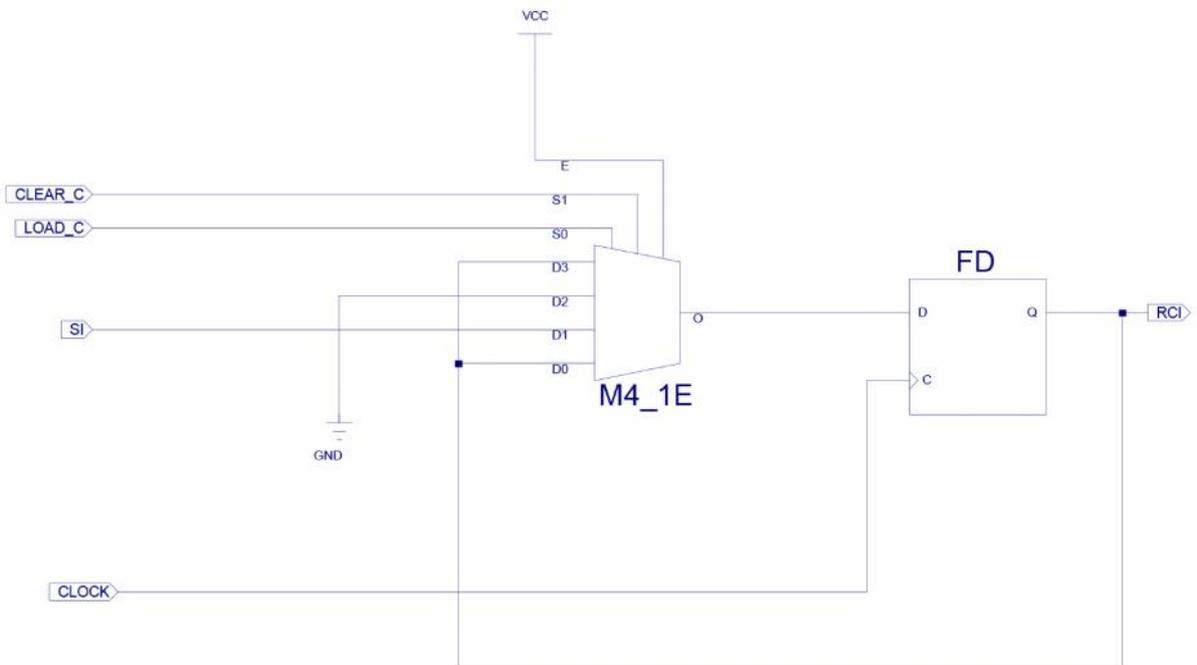


3.2 Módulo Somador

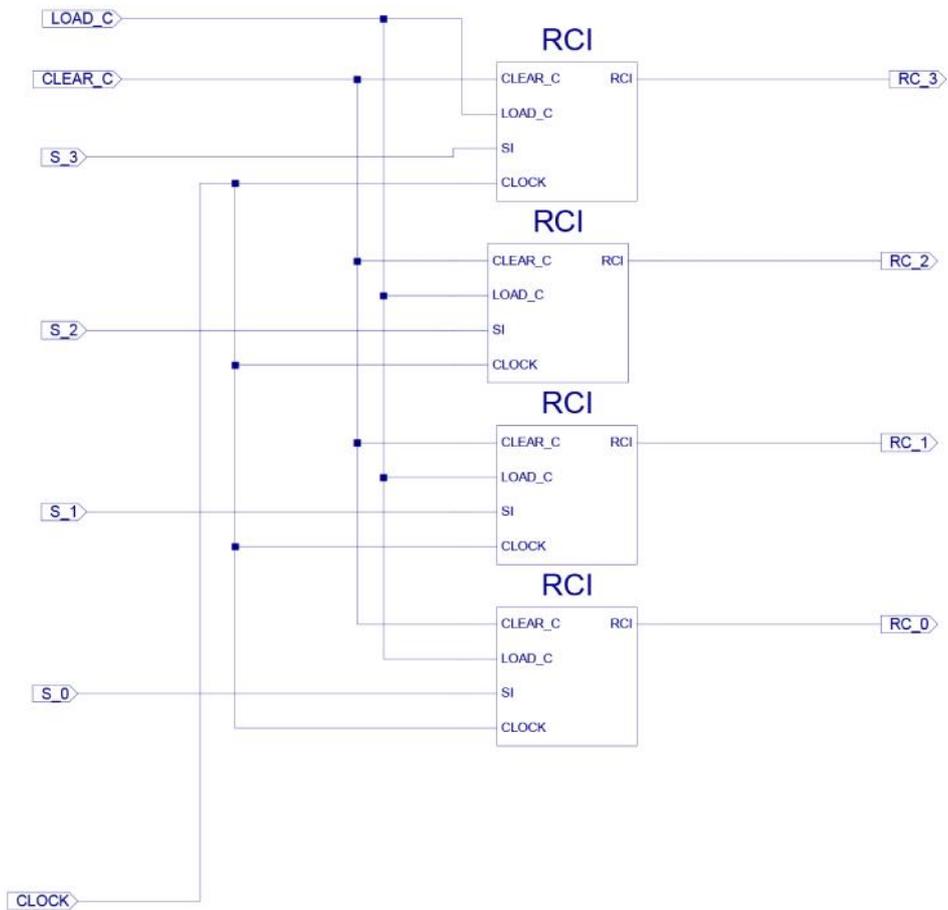


3.3 Módulo REG_C

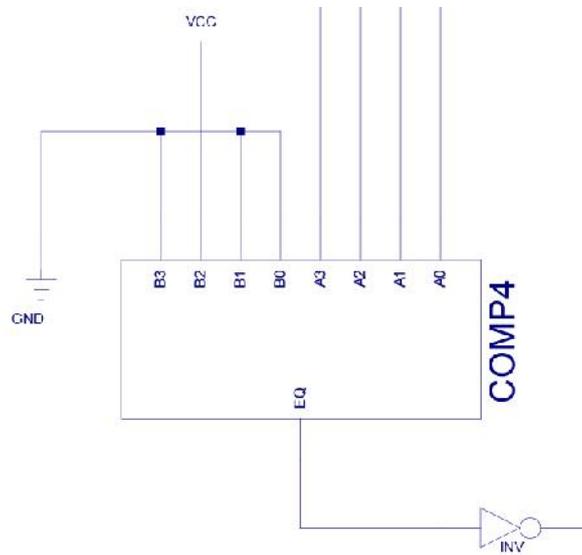
RCI



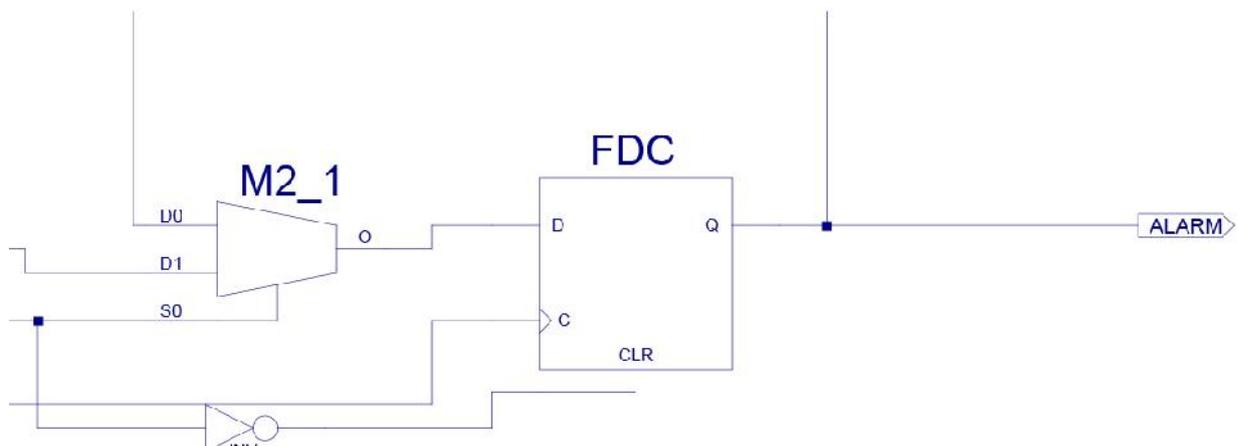
RC



3.4 Módulo OPER K₃

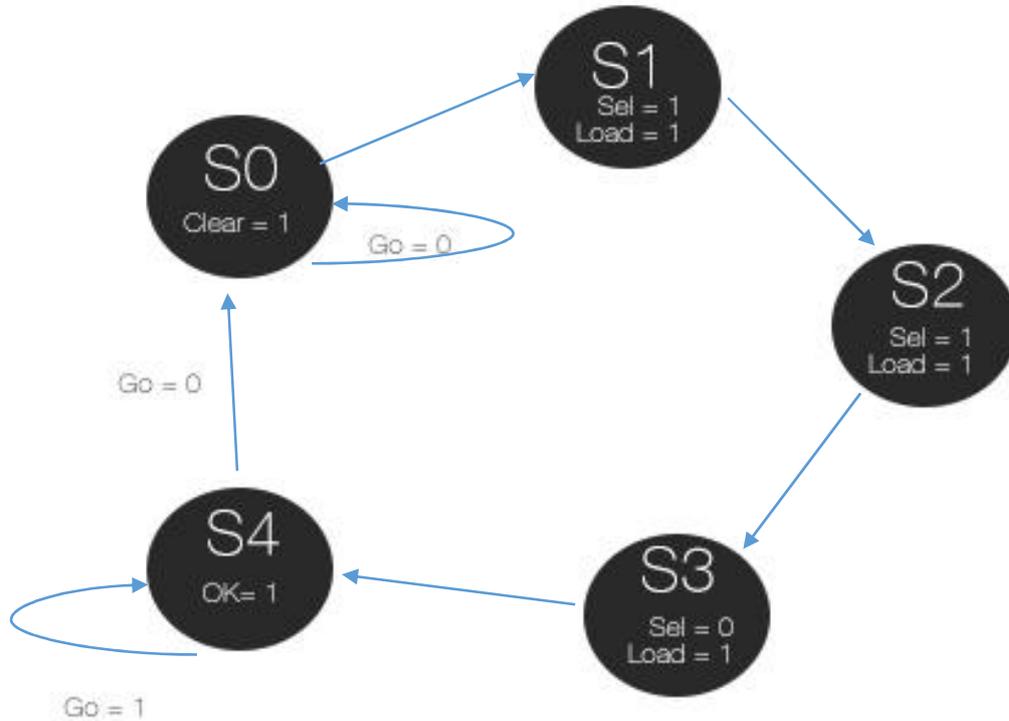


3.5 Módulo F



4 Síntese – Parte de Controlo

4.1 Diagrama de Estados



4.2 Tabela de Transição de Estados

| | | |
|--------|--------|--------|
| Go = 0 | S0 000 | S0 000 |
| | S1 001 | S1 001 |
| | S2 010 | S2 010 |
| | S3 011 | S3 011 |
| | S4 100 | S4 100 |
| Go = 1 | S0 000 | S1 001 |
| | S1 001 | S2 010 |
| | S2 010 | S3 011 |
| | S3 011 | S4 100 |
| | S4 100 | S4 100 |

4.3 Codificação de Estados

| | |
|----|-----|
| S0 | 000 |
| S1 | 001 |
| S2 | 010 |
| S3 | 011 |
| S4 | 100 |

4.5 Expressões simplificadas das saídas por Mapas de Karnaugh

$$Clear = \overline{Q2} \cdot \overline{Q1} \cdot \overline{Q0}$$

| | | | | |
|----------------|---|---|---|---|
| GO Q2 Q1 Q0 | 1 | 0 | 0 | 1 |
| | 0 | X | X | 0 |
| | 0 | X | X | 0 |
| | 0 | X | X | 0 |

$$Load = Q0 + Q1$$

| | | | | |
|----------------|---|---|---|---|
| GO Q2 Q1 Q0 | 0 | 0 | 0 | 0 |
| | 1 | X | X | 1 |
| | 1 | X | X | 1 |
| | 1 | X | X | 1 |

$$\text{Select} = Q0 \oplus Q1$$

~~GO Q2~~
Q1 Q0

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | X | X | 1 |
| 0 | X | X | 0 |
| 1 | X | X | 1 |

$$OK = Q2$$

~~GO Q2~~
Q1 Q0

| | | | |
|---|---|---|---|
| 0 | 1 | 1 | 0 |
| 0 | X | X | 0 |
| 0 | X | X | 0 |
| 0 | X | X | 0 |

4.6 Expressões simplificadas das entradas dos Flip-flops por Mapas de Karnaugh

$$J_2 = \overline{G_0} \cdot Q_0 \cdot Q_1$$

GO Q2
Q1 Q0

| | | | |
|---|---|---|---|
| 0 | X | X | 0 |
| 0 | X | X | 0 |
| 0 | X | X | 1 |
| 0 | x | X | X |

$$K_2 = \overline{G_0}$$

GO Q2
Q1 Q0

| | | | |
|---|---|---|---|
| X | 1 | 0 | X |
| X | X | X | X |
| X | X | X | X |
| X | X | X | X |

$$J1 = GO . Q0$$

| | | | | |
|------------------|---|---|---|---|
| GO Q2 | | | | |
| Q1 Q0 | | | | |
| | 0 | 0 | 0 | 0 |
| | 0 | X | X | 1 |
| | X | X | X | X |
| | X | X | X | X |

$$K1 = GO . Q0$$

| | | | | |
|------------------|---|---|---|---|
| GO Q2 | | | | |
| Q1 Q0 | | | | |
| | X | X | X | X |
| | X | X | X | X |
| | 0 | X | X | 1 |
| | 0 | X | X | 0 |

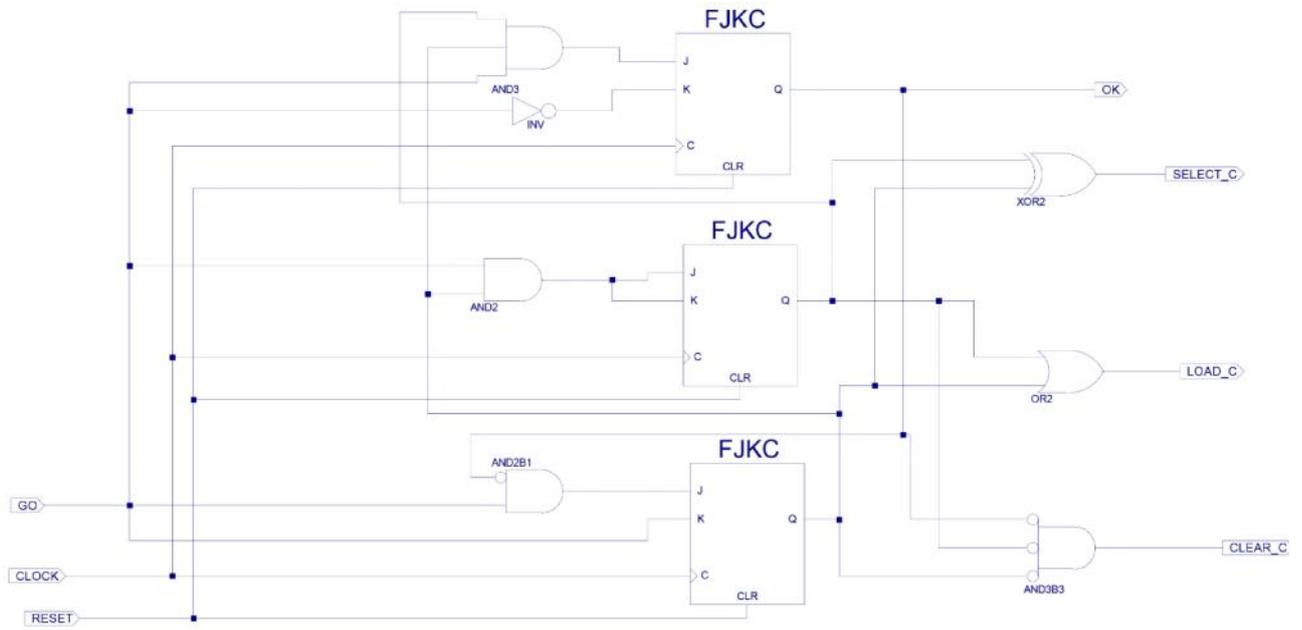
$$J0 = G0 \cdot \overline{Q2}$$

| | | | | |
|------------------|----|---|---|---|
| GO Q2 | | | | |
| Q1 | Q0 | | | |
| | 0 | 0 | 0 | 1 |
| | X | X | X | X |
| | X | X | X | X |
| | 0 | X | X | 1 |

$$K0 = G0$$

| | | | | |
|------------------|----|---|---|---|
| GO Q2 | | | | |
| Q1 | Q0 | | | |
| | X | X | X | X |
| | 0 | X | X | 1 |
| | 0 | X | X | X |
| | X | X | X | 1 |

4.7 Esquemático da Parte de Controlo



5 Implementação na FPGA Spartan 3E

```
1 net "A0" loc= "K3";
2 net "A1" loc= "B4";
3 net "B0" loc= "P11";
4 net "B1" loc= "L3";
5 net "GO" loc= "N3";
6 net "CLOCK" loc= "A7";
7 net "CLOCK" CLOCK_DEDICATED_ROUTE = FALSE;
8 net "RESET" loc= "C11";
9 net "ALARME" loc= "M5";
10
```

6 Validação através de simulações

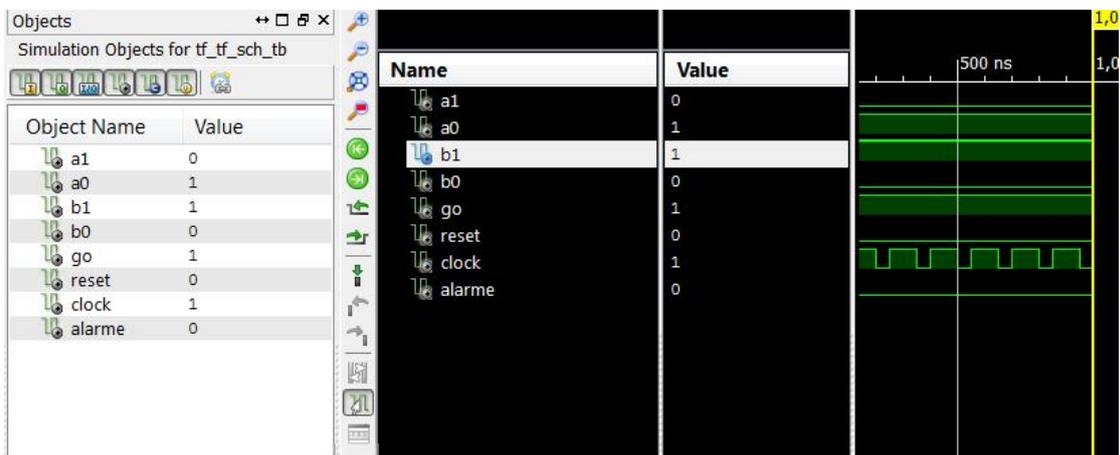
6.1 Definição das situações para teste

Na elaboração deste projecto, antes de passar à implementação na FPGA Spartan 3E, foram feitos vários testes em ambiente de simulação para garantir o correcto funcionamento da solução. Aqui são demonstrados dois casos onde o valor do alarme é diferente.

Resultados das simulações

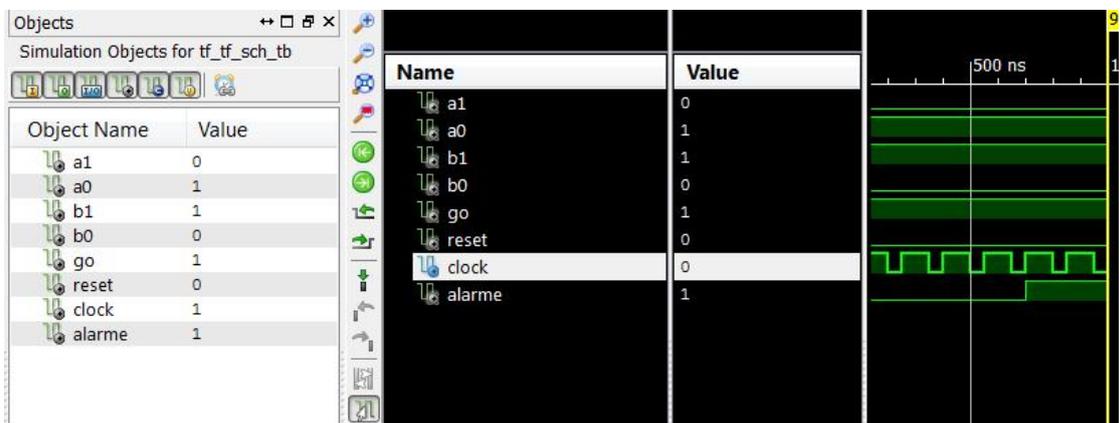
Simulação com $A = 1$ e $B = 2$ ($2 * 1 + 1 * 2 = 4$)

Daqui resulta que o valor da expressão $2 * A + 1 * B$ é igual a 4, pelo que o alarme não é activado.



Simulação com $A = 3$ e $B = 3$ ($2 * 3 + 1 * 3 = 9$)

Temos então que o resultado da equação é diferente da constante $k3$, sendo que neste caso temos a activação do alarme.



7 Resultados experimentais

Ao implementar a solução a nível dos componentes, foram efectuados vários testes em ambiente de simulação no xlinx de modo a compreender se o comportamento do sistema de alarme era o esperado. Assim foram testados vários cenários, onde o alarme pudesse ser activado, e onde não é expectável a sua activação.

Após todos estes testes procedeu-se à implementação da solução na FPGA Spartan 3E e foi testado o seu comportamento em comparação com as simulações efectuadas no xlinx. Ao que se verificou que o comportamento é o correcto.

Quanto á definição do dos inputs na placa, foram definidos os switchs para o Go, A0, A1, B0 e B1; e dois botões para o reset e para o clock. Quanto ao output foi apenas utilizado um led para simbolizar a ocorrência de um alarme.

8 Conclusões e Observações

Com o desenvolvimento deste trabalho é possível concluir que no que diz respeito à carga horaria esta foi adequada, tendo em conta os moldes no qual este trabalho é inserido no programa das aulas praticas.

Pode-se também referir que para além de alguma facilidade em compreender os conceitos base para o desenvolvimento do projecto, o software no qual foi desenvolvido o projecto apresentou se como uma das grandes dificuldades, uma vez que a sua interface é confusa e pouco funcional e apresenta alguns bugs que inserem incorrecções na implementação.

Existe algum espaço para a optimização, mas devido à dimensão deste projecto considera-se que esta não faria uma grande diferença, apenas seria rentável com a escalabilidade da implementação.