

Ricardo Alexandre Ventura Matos

43321

3º Trabalho Prático

Samuel Sampaio Costa

43552

14-01-2017

Pedro Miguel Carvalho Rocha

43884

Grupo 4

Introdução

O presente relatório, referente ao terceiro de três trabalhos práticos (Identificador de Portas Lógicas), acompanha a realização do projeto de um sistema de identificação e teste de portas lógicas OR, AND e XOR de duas entradas, em que o sistema verifica se a porta lógica em avaliação implementa uma das funções mencionadas, caso contrário sinaliza a existência de erro.

O sistema tem o diagrama de blocos apresentado na Figura 1, constituído pelos seguintes elementos:

- um botão B para dar início à operação de identificação da porta lógica;
- duas saídas GA e GB, que estabelecem as várias combinações nas entradas da porta lógica em avaliação no sentido de determinar a função lógica.
- uma entrada GOut, para verificar qual o valor lógico da saída da porta lógica em avaliação para cada uma das combinações das entradas (GA e GB).

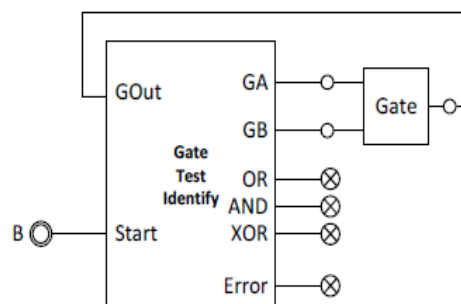


Figura 1- Sistema de identificação e teste de portas lógicas

Modo de funcionamento

- A identificação da porta lógica inicia-se estabelecendo a ligação elétrica entre os pinos da porta em avaliação e os sinais GA e GB e GOut.
- Pressiona-se o botão B, para dar início ao processo de identificação e aguarda-se que o sistema indique qual a função lógica desempenhada pela porta em avaliação, ou caso não coincida com nenhuma das funções estabelecidas ativa a saída Error. No início do processo de identificação todas as saídas são colocadas a zero. A indicação de qual a função lógica identificada mantém-se ativa até novo processo de identificação.

Arquitetura proposta

Na Figura 2 apresenta-se o diagrama de blocos de uma arquitetura para o sistema. A implementação recorre ao módulo examinador realizado no âmbito do 2º trabalho prático.

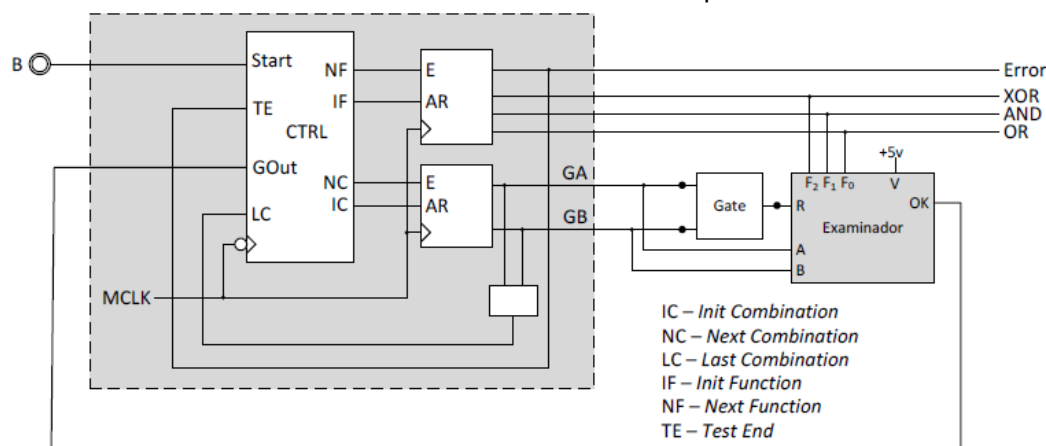


Figura 2

Sequencia de trabalho

O presente trabalho foi realizado em três fases:

1ª fase: conceção da máquina de estado que implementa o modulo CTRL e seu ASM-chart. Implementação na PAL v750c recorrendo a flip-flops do tipo D internos, e implementação da lógica combinatória geradora de estado seguinte e sinais de saída. As entradas e saídas do módulo foram simuladas recorrendo respetivamente a comutadores e LEDs da base de ensaio ATB.

2ª fase: nova implementação do ASM-chart da 1ª fase na PAL v750c, mas desta vez recorrendo a sequence/present.

3ª fase: sobre a implementação da 2ª fase foram implementados três outros módulos funcionais, tendo o projeto evoluído para a implementação e integração na PAL v750c do modulo examinador realizado no âmbito do 2º trabalho prático.

1ª fase

Modulo CTRL

O modulo CTRL é onde se encontra o núcleo da máquina de identificação de portas (funções). Para a sua implementação foi elaborada uma máquina de estados representada pelo ASM-chart da figura 3.

O processo de identificação da função passa por testar todas as combinações possíveis de cada função. O teste termina (TE=1) quando todas as funções tiverem sido testadas sem encontrar uma válida, ou quando for encontrada uma função válida.

Uma função é válida quando todas as combinações possíveis da função tiverem sido testadas e forem válidas, isto é, desde o inicio das combinações (IC), percorrendo todas as combinações possíveis (NC), até á ultima (LC) estas tiverem sido testadas pelo examinador com indicação que são validas (Gout=1). Se o examinador indica que uma das combinações não é válida (Gout=0), a função não é válida.

Ao ligar a máquina esta inicia o funcionamento no estado 0 ('b'000), e transitará para o estado 1 ('b'001) se o utilizador acionar Start (S=1).

No estado 1 a máquina ativa a saída IF (IFunc), que representa o início de função, isto é, indica que a maquina inicia o processo de verificação de funções, e sem mais condições passa para o estado 2 ('b' 010).

No estado 2 ativa a saída NF (NFunc), que indica que a maquina seleciona uma nova função, e ativa a saída IC (IComb), que indica que a maquina vai testar a primeira combinação. Se, nesse estado, TE estiver ativado (TE=1), isto é, a máquina fornecer indicação de que todas as funções foram testadas sem encontrar uma válida, a maquina passa para o estado 4 ('b' 100). Por outro lado, se TE não esta ativo (TE=0), é verificada a primeira combinação. Se esta não for válida (Gout=0) mantém o estado, ie, é testada uma nova função. Se aquela for válida (Gout=1) passa para o estado 3 ('b' 011).

No estado 3, ativa a saída NC (NComb), que indica que a máquina seleciona uma nova combinação. Se esta combinação não for válida (Gout=0) passa para o estado 2, para selecionar uma nova função e iniciar as combinações. Se a combinação é valida (Gout=1), verifica se se trata da ultima combinação. Se não for (LC=0) mantém o estado. Se se tratar da última combinação (LC=1) a maquina passa para o estado 4 ('b' 100), o que significa que testou todas as combinações da função com resultado positivo, mostrando que função é que a porta testada implementa.

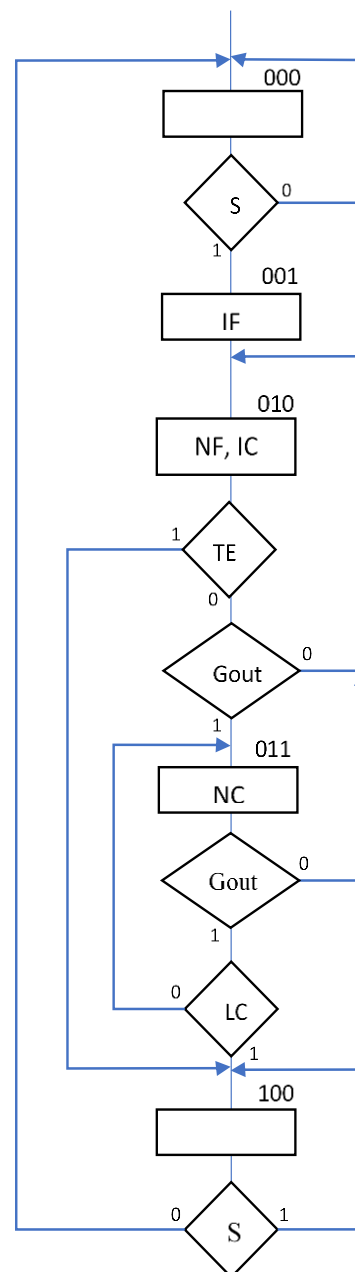


Figura 3- ASM-chart

A máquina mantém-se no estado 4 enquanto Start estiver ativo ($S=1$), passando para o estado 0 ('b' 000) quando Start deixar de estar ativo ($S=0$).

Implementação modulo CTRL

Para implementar a máquina de estados representada pelo ASM chart da Figura 3 foram necessários 3 flip-flop D [X0..2], ligados a 3 multiplexer de 3 bits para ser obtida a lógica do estado seguinte, conforme a figura 4.

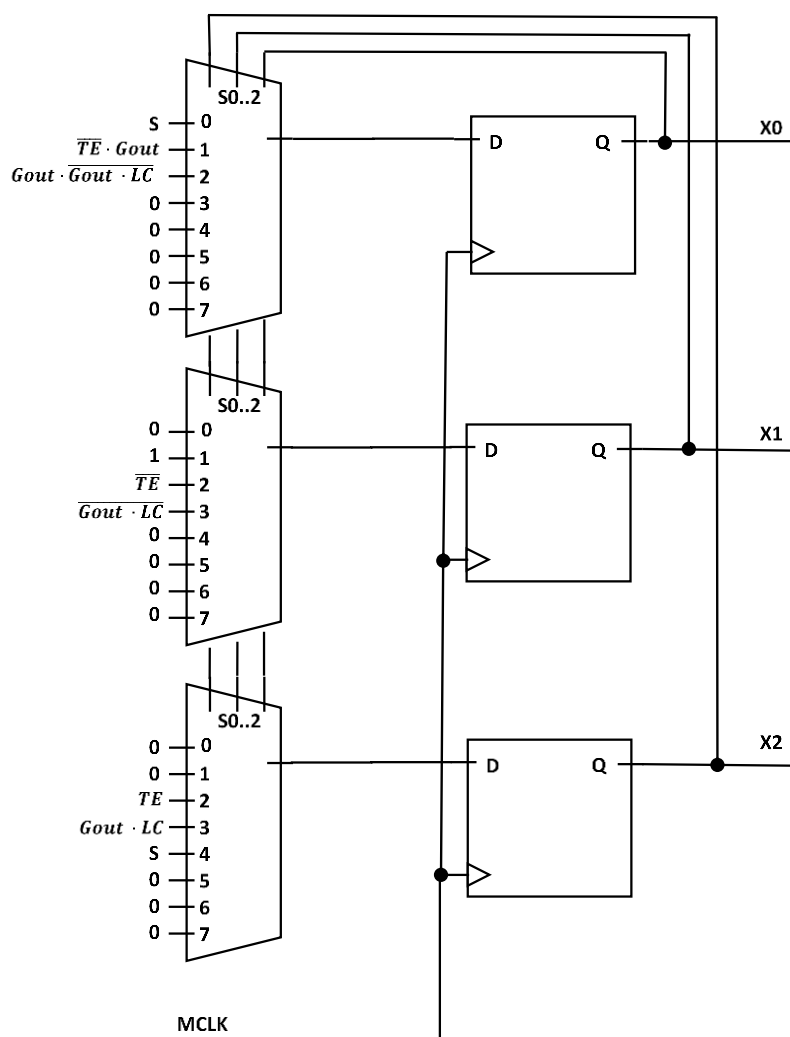


Figura 4- Esquema implementação ASM com multiplexer

As saídas são dadas pelas seguintes expressões:

$$IFunc = IF = \overline{X2} \cdot \overline{X1} \cdot X0$$

$$NFunc = NF = \overline{X2} \cdot X1 \cdot \overline{X0}$$

$$IComb = IC = \overline{X2} \cdot X1 \cdot \overline{X0}$$

$$NFunc = NC = \overline{X2} \cdot X1 \cdot X0$$

De reparar que $NF=IC$, pois de acordo com a máquina de estados concebida, NF e IC só estão ativos enquanto a máquina estiver no estado 2 e em mais nenhuma circunstância.

A figura 5 apresenta o código implementado na ALU para esta fase.

```

Name          IDENTIFICADOR_PORTAS LOGICAS - Fase 1;
PartNo        00 ;
Date          15-12-2016 ;
Revision      01 ;
Designer      PR RM SC ;
Company       CCISEL ;
Assembly      None ;
Location      ;
Device        v750c ;

/***** INPUT PINS *****/
PIN 1 = MCLK;
PIN 2 = Start;
PIN 3 = TE;
PIN 4 = Gout;
PIN 5 = LC;
/***** OUTPUT PINS *****/
PIN 23 = X0;
PIN 22 = X1;
PIN 21 = X2;
PIN 20 = NFunc;
PIN 19 = IFunc;
PIN 18 = NComb;
PIN 17 = IComb;

[X0..2].AR = 'b' 0;
[X0..2].SP = 'b' 0;
[X0..2].CK = MCLK;

X0.D = !X2 & !X1 & !X0 & Start
# !X2 & X1 & !X0 & !TE & Gout
# !X2 & X1 & X0 & Gout & !(Gout & LC);

X1.D=  !X2 & !X1 & X0
#  !X2 & X1 & !X0 & !TE
#  !X2 & X1 & X0 & !(Gout & LC);

X2.D=  !X2 & X1 & !X0 & TE
# !X2 & X1 & X0 & Gout & LC
#  X2 & !X1 & !X0 & Start;

IFunc = !X2 & !X1 & X0;
NFunc = !X2 & X1 & !X0;
IComb = !X2 & X1 & !X0;
NComb = !X2 & X1 & X0;

```

Figura 5- Código ALU - Multiplexer



2ª fase

Nesta fase, pretendeu-se implementar a máquina de estados concebida recorrendo a sequence/present, conforme a Figura 6.

```
Name          IDENTIFICADOR_PORTAS LOGICAS - Fase 2;
PartNo         00 ;
Date           15-12-2016 ;
Revision       01 ;
Designer       PR RM SC ;
Company        CCISEL ;
Assembly       None ;
Location       ;
Device         v750c ;

/***** INPUT PINS *****/
PIN 1 = MCLK;
PIN 2 = Start;
PIN 3 = TE;
PIN 4 = Gout;
PIN 5 = LC;
/***** OUTPUT PINS *****/
PIN 23 = X0;
PIN 22 = X1;
PIN 21 = X2;
PIN 20 = NFunc;
PIN 19 = IFunc;
PIN 18 = NComb;
PIN 17 = IComb;

[X0..2].AR = 'b' 0;
[X0..2].SP = 'b' 0;
[X0..2].CK = MCLK;

SEQUENCE [X2,X1,X0] {

PRESENT 'b' 000
    IF Start NEXT 'b' 001;
    DEFAULT NEXT 'b' 000;

PRESENT 'b' 001
    OUT IFunc;
    DEFAULT NEXT 'b' 010;

PRESENT 'b' 010
    OUT NFunc, IComb;
    IF TE NEXT 'b' 100;
    IF !TE & Gout NEXT 'b' 011;
    DEFAULT NEXT 'b' 010;

PRESENT 'b' 011
    OUT NComb;
    IF !Gout NEXT 'b' 010;
    IF Gout & LC NEXT 'b' 100;
    DEFAULT NEXT 'b' 011;

PRESENT 'b' 100
    IF !Start NEXT 'b' 000;
    DEFAULT NEXT 'b' 100;

}
```

Figura 6- Código PAL – Sequence Present

3ª fase

Na 3ª fase do projeto pretendeu-se desenvolver dois módulos, adaptar o Examinador desenvolvido a propósito do 2º TP e interligar esses 3 módulos por forma a obter um Identificador de Portas Lógicas. Segue-se uma breve apresentação da função de cada módulo, sua estrutura e integração. Assim, os três módulos desenvolvidos foram:

- Módulo Select Function;
- Módulo Select Combination;
- Adaptação e Integração do Módulo Examinador.

Módulo Select Function

No módulo seletor de função (Select Function), optou-se por utilizar uma variação do método Shift Right a 4 bits, em que o bit à esquerda é transferido para a direita (shift right) em função de NC.

Assim sendo, o Shift Right desenvolvido é composto por 4 flip-flop do tipo D em que o estado seguinte é determinado pelo flip-flop anterior quando NC está ativado (NC=1), caso contrário o estado é mantido. O primeiro flip-flop recebe 'b' 1 quando NC está ativo, por não ter outro flip-flop a montante.

Para este método, respeitaram-se as prioridades das operações booleanas. Cada um dos primeiros 3 bits da esquerda do Shift Right corresponde a uma função booleana. À esquerda temos a menos prioritária, o XOR (1000), que é a primeira função a ser testada. De seguida, o OR (1100) e o AND (1110). Por final, o error (1111) é gerado quando NC for ativado e a palavra binária 1110 estiver no Shift Right (AND). As saídas deste módulo (SR0, SR1, SR2, SR3) são entradas do módulo examinador.

Quando SR3 for ativado quer dizer que se chegou ao fim do teste (TE). Por outro lado, se SR3 for ativado, a porta em teste não corresponde a nenhuma das três funções. SR3 é, portanto, uma entrada do módulo CTRL, uma entrada do examinador e uma saída LED na placa de teste ATB.

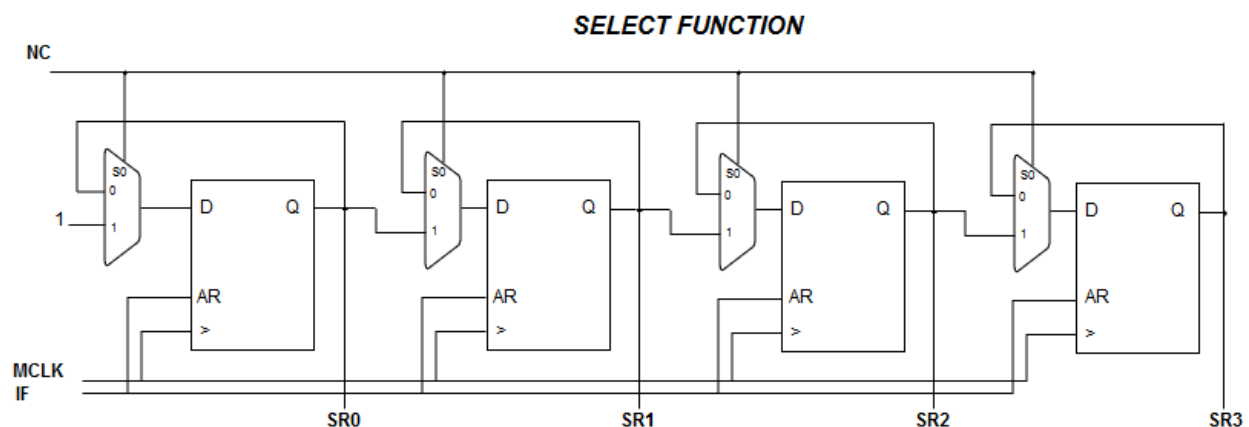


Figura 7- Select function

Módulo Select Combination

O módulo select combination é um contador simples de dois bits implementado à custa de dois flip-flops do tipo D interligados. Aquele recebe como entradas NC (NComb) e IC (IComb) e gera duas saídas C0 e C1, bem como um sinal LC.

Foram obtidas as seguintes expressões referentes à lógica do estado seguinte na geração dos sinais C0 e C1: $C0.D = NComb \oplus C0$; $C1.D = C0.NComb + C1$.

Como IC está ligado como reset assíncrono nos dois flip-flops, assim que este esteja ativo, e independentemente do clock, as duas saídas C0 e C1 são levadas a 0.

Neste caso, NC funciona como enable, isto é, sempre que este esteja ativo, o contador é incrementado em uma unidade ao impulso de um clock.

Conforme explicitado no diagrama de blocos, C0 e C1 correspondem aos sinais A e B que dão entrada no módulo Examinador, ie a combinação a ser testada. Para além disso, C0 e C1 são duas saídas da PAL, ligadas a entradas de "Gate", a porta lógica a ser testada

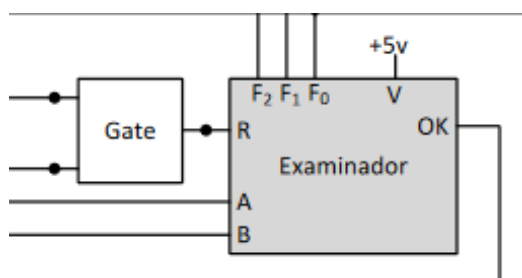
Finalmente, o sinal LC sinaliza que se trata da última combinação ($C0=1$ e $C1=1$). Assim: $LC = C0.C1$

Adaptação e Integração do Módulo Examinador

Fizeram-se corresponder às entradas do examinador, saídas do módulo select combination e select function da seguinte forma (Figura 8):

Saídas do módulo
Select Combination:

F0 = SR2
F1 = SR1
F2 = SR0



Saídas do módulo
Select Function:

A = C0
B = C1

Figura 8- integração examinador

É de salientar que a saída OK do examinador não foi mantida, tendo sido substituída pela saída RESULT. A figura 9 apresenta o diagrama de blocos final, a figura 10 representa o código implementado na PAL para esta fase, e a figura 11 representa o PIN-OUT da montagem na PAL v750c

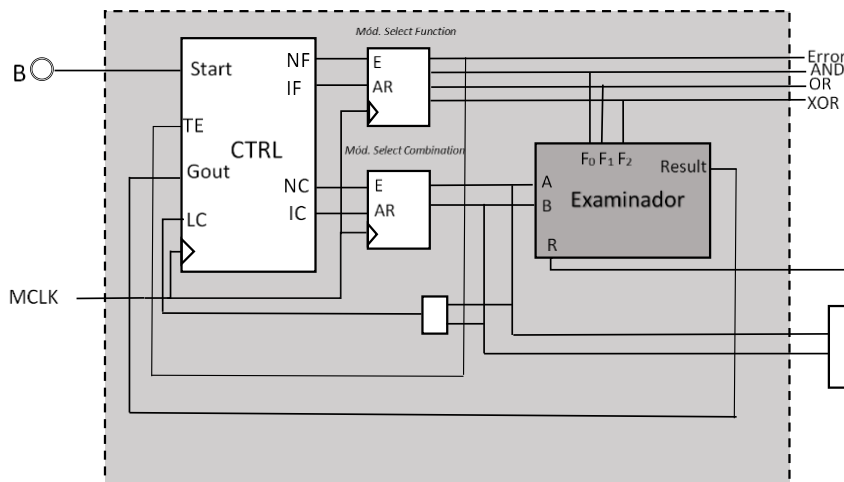


Figura 9- Diagrama de blocos final

```

Name          IDENTIFICADOR_PORTAS LOGICAS ;
PartNo        00 ;
Date          15-12-2016 ;
Revision      01 ;
Designer      PR RM SC ;
Company       CCISEL ;
Assembly      None ;
Location      ;
Device        v750c ;

/***** INPUT PINS *****/
PIN 1 = MCLK;
PIN 2 = Start;
PIN 3 = R;
/***** OUTPUT PINS *****/
PIN 23 = SR0;
PIN 22 = SR1;
PIN 21 = SR2;
PIN 20 = SR3;
PIN 19 = X0;
PIN 18 = X1;
PIN 17 = X2;
PIN 16 = Gout;
PIN 15 = C0;
PIN 14 = C1;

[X0..2].CK = !MCLK;
[X0..2].AR = 'b' 0;
[X0..2].SP = 'b' 0;

SEQUENCE [X2,X1,X0] {
PRESENT 'b' 000
    IF Start NEXT 'b' 001;
    DEFAULT NEXT 'b' 000;
PRESENT 'b' 001
    OUT IFunc;
    DEFAULT NEXT 'b' 010;
PRESENT 'b' 010
    OUT NFunc, IComb;
    IF TE NEXT 'b' 100;
    IF !TE & Gout NEXT 'b' 011;
    DEFAULT NEXT 'b' 010;
PRESENT 'b' 011
    OUT NComb;
    IF !Gout NEXT 'b' 010;
    IF Gout & LC NEXT 'b' 100;
    DEFAULT NEXT 'b' 011;
PRESENT 'b' 100
    IF !Start NEXT 'b' 000;
    DEFAULT NEXT 'b' 100;
}

/***** SELECT FUNCTION *****/
[SR0..3].CK = MCLK;
[SR0..3].AR = IFunc;
[SR0..3].SP = 'b' 0;

SR0.D=NFunc & 'b' 1 # !NFunc & SR0;
SR1.D=NFunc & SR0 # !NFunc & SR1;
SR2.D=NFunc & SR1 # !NFunc & SR2;
SR3.D=NFunc & SR2 # !NFunc & SR3;

TE = SR3;

/***** SELECT COMBINATION*****/
[C0..1].CK=MCLK;
[C0..1].AR=IComb;
[C0..1].SP = 'b' 0;

C0.D = NComb $ C0;
C1.D = C0 & NComb # C1;

LC = C0 & C1;

```



```

/***** PRIORITY *****/

CF0 = SR2 # !SR2 & !SR1 & SR0 ;
CF1 = !SR2 & SR0 # SR1 & !SR2;

NF = SR3;

/***** FUNCTIONRESULT *****/

FR = CF1 & !CF0      & C1 #
      CF1            & C0 & !C1 #
      !CF1 & CF0 & C0 & C1 #
      CF1 & CF0 & !C0 & C1;

/***** RESULT *****/
Gout = !(FR$R);

```

Figura 10- Código PAL –final – 3ª fase

PIN – OUT

MCLK	I1	1	24	5V	VCC
Start	I2	2	23	I010	SR0
R	I3	3	22	I09	SR1
	I4	4	21	I08	SR2
	I5	5	20	I07	SR3
	I6	6	19	I06	X0
	I7	7	18	I05	X1
	I8	8	17	I04	X2
	I9	9	16	I03	Gout
	I10	10	15	I02	C0
	I11	11	14	I01	C1
GROUND	GND	12	13	I12	

Esquema de Ligações na PAL

Figura 11 – PIN-OUT da montagem

Conclusões

O projeto do Identificador de portas lógicas foi implementado e montado em sala de aula. Os resultados obtidos encontram-se em conformidade com o enunciado. As saídas Error e/ou os indicadores de função permanecem ativos quando o teste termina e até o utilizador voltar a ativar o Start.

São de salientar duas conclusões, uma referente à utilização de clock em projetos modulares, e outra referente à PAL v750c:

- (i) Como existem flip-flop nos módulos Select Combination e Select Function cujo sinal depende da informação de outros no módulo CTRL, surgiu a necessidade de o clock do módulo CTRL estar em fase distinta dos clock dos restantes módulos, caso contrário os módulos Select Function e Select Combination receberiam informação relativa ao estado anterior do módulo CTRL.
- (ii) Devido às características da PAL v750c, escolheram-se os pinos de saída em função do número de termos máximos suportados por cada pino.