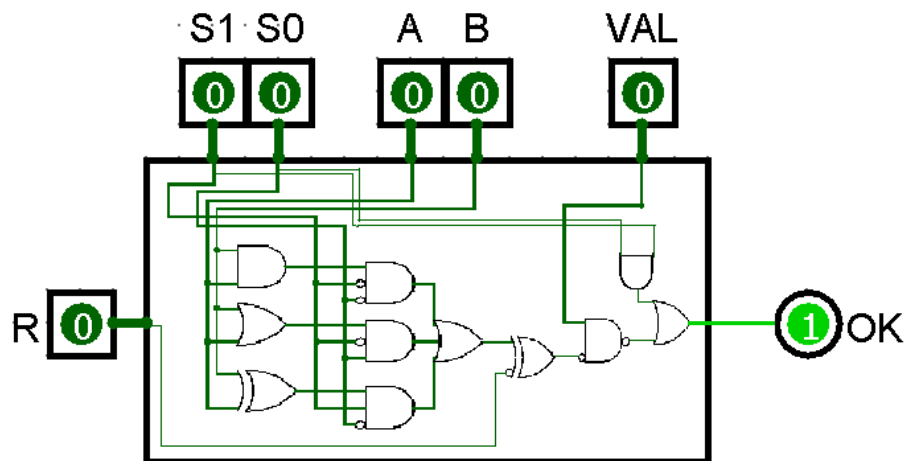


Lógica e Sistemas Digitais

Tester



Realizado pelo grupo 8:

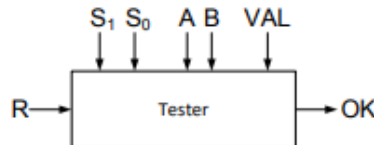
Paulo Rosa	44873
Gonçalo Santos	44587
Diogo Gouveia	44883

Docente: José Paraizo

29/3/18

Enunciado

Neste primeiro trabalho pretende-se que os alunos projectem e implementem um sistema para apoio à avaliação da disciplina de Lógica e Sistemas Digitais no capítulo da Álgebra de Boole. O objectivo do sistema é suportar a realização da avaliação por um examinador que não domine necessariamente o tema. Na figura abaixo é apresentado o diagrama de blocos do sistema.



Neste exame, que verifica os conhecimentos acerca de funções lógicas de duas variáveis, participam um examinador e um examinando.

O exame tem a seguinte sequência:

1. O examinador através de dois bits (**S1**,**S0**) selecciona uma de três funções lógicas de duas variáveis e estabelece através das entradas **A** e **B** o valor lógico atribuído a cada uma das entradas da função;
2. De seguida, indica ao examinando a função que seleccionou e quais os valores de **A** e de **B**;
3. Aguarda que o examinando, dada a função seleccionada e os valores lógicos presentes em **A** e **B**, estabeleça na entrada **R** o valor lógico que corresponde ao resultado da função;
4. Estabelecido o valor de **R**, o examinador actua a entrada **VAL** permitindo observar na saída **OK** a correcção da resposta.

As funções a avaliar serão as seguintes: AND, OR e XOR.

A relação entre a combinação de **S0** e **S1** com a função seleccionada é definida pelo projectista. O sistema não deverá ser sensível às entradas A, B, R e VAL caso **S0** e **S1** não estejam a seleccionar uma das três operações.

A saída **OK** deverá permanecer ao valor lógico 1 enquanto o examinando não indicar a resposta, ou seja, enquanto a entrada **VAL** estiver inactiva. Quando a entrada **VAL** for activada, a saída **OK** reage em lógica positiva, ou seja, toma o valor lógico 1 se a resposta estiver correcta.

Apresente uma solução para o problema, que deverá ter uma estrutura modular baseada em blocos funcionais, e implemente-a. Especifique em diagrama de blocos a interface entre os vários módulos, designadamente indicando a natureza de cada sinal quanto a ser activo a 1 ou a 0.

1. Introdução:

Dividimos a nossa solução do circuito em 2 partes em logisim. O circuito chamado “SpecialMux” e o circuito “Okounao” que juntos formam a Tester.

O “SpecialMux” trata das diferentes configurações dos inputs S0, S1, A e B(a questão dada) para dar resultado a um certo *output*(que representa a solução certa com as dadas configurações).

O “Okounao” usa o resultado do *output* do circuito anterior em conjunção com a resposta do aluno(R). E usa este *output* mais a indicação pelo examinando para mostrar a validação da questão dada(VAL) a fim de mostrar se a resposta do aluno está certa ou não(o *output* OK).

Estes dois circuitos são complementados para formarem o circuito desejado, nomeado de “main”. O Tester.

2. Realização do projecto:

2.1 SpecialMux:

Como entrada para todas as portas lógicas, temos o A e B. Estas portas são o And, Or e Xor. Destas diferentes operações, usamos o S0 e S1 para definir qual delas desejamos usar.

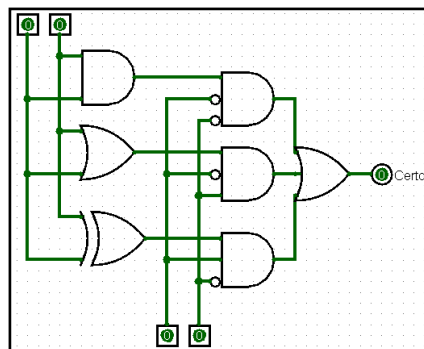
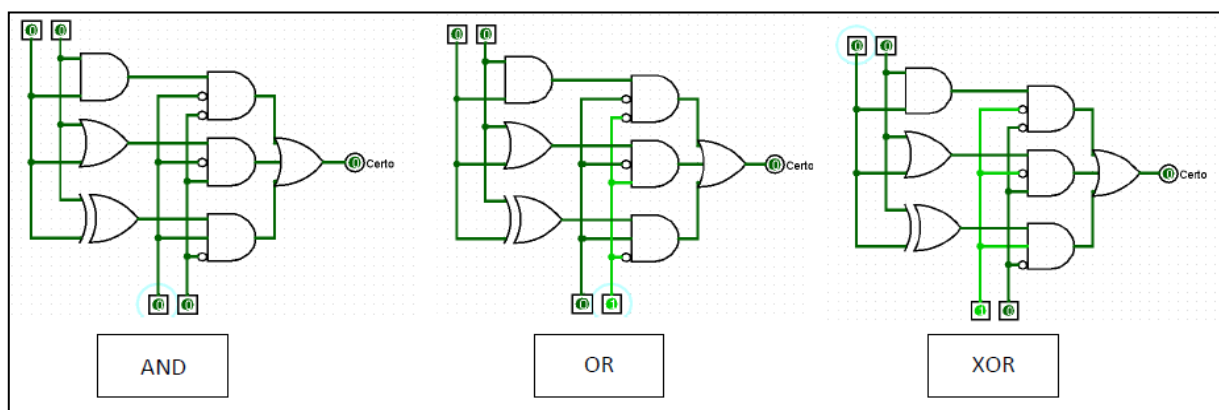
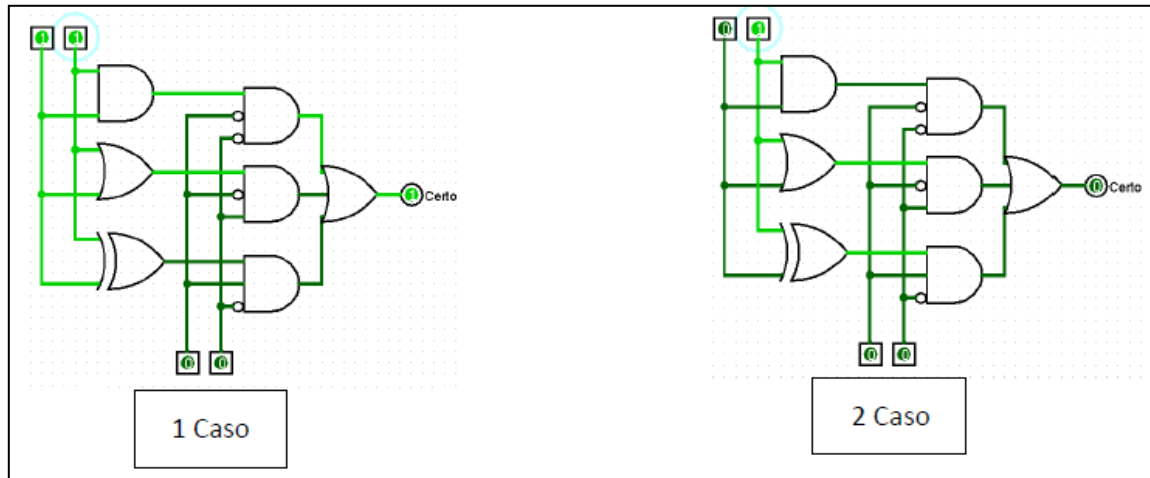


Fig 1: SpecialMux

Para usar as expressões AND, OR e XOR, usamos para S0 e S1, respectivamente, 00, 01 e 10. E a explicação para a configuração 11 é tratada em 2.2 Okounao.



Abaixo, demonstramos um exemplo em que desejamos avaliar $A \cdot B$. Para tal, selecionamos 00 nos seletores S1 e S0. Para obter uma demonstração verdadeira, iremos meter A a 1 e B a 1 (1º Caso), e para uma demonstração falsa A a 0 e B a 1 (2º Caso)



2.2 Okounao:

Depois de já termos definido o nosso circuito que dá a resposta correta (verdadeiro ou falso, 1 ou 0), comparamos este valor com a resposta do aluno (usando um XOR com uma negação) que resulta na validação da resposta do aluno, ou seja, se a resposta do aluno está certa ou errada.

No entanto, só depois de ser acionado o VAL pelo examinador, veremos se a resposta é válida ou não. A variável de saída OK é somente 0 se R e R do aluno forem diferentes uma da outra e se VAL for verdadeiro.

Quando o S1 e S0 estiverem ambos a 1, usamos um OR que, sendo uma porta “absorvente”, causará o OK a permanecer a 1, e sendo este *output* independente de a resposta do aluno estar certa e do VAL estar ligado ou não, é por este motivo que é esta porta é posicionada à frente destas validações.

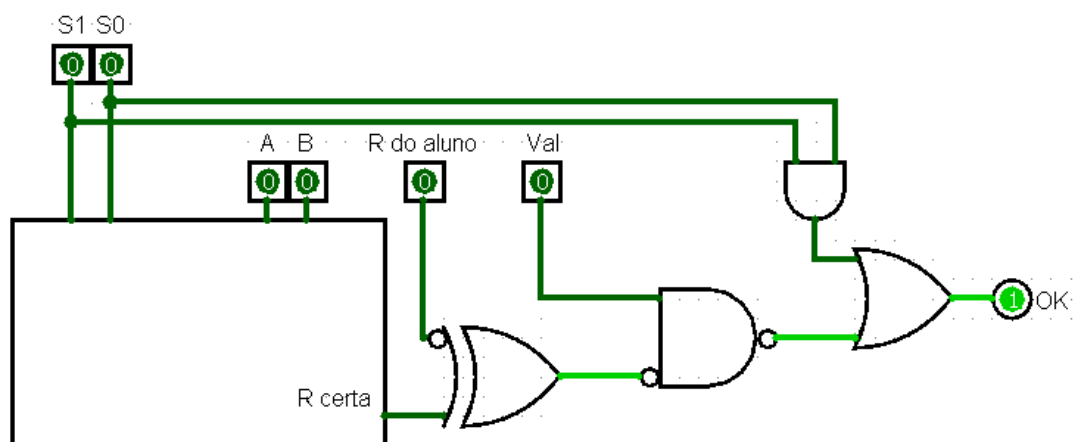


Fig 2: Okounao

2.3 Código do problema em CUPL:

Portas {

Variáveis intermédias →

```
Device v750c ;

/* ***** INPUT PINS ***** */
PIN 1 = A ; /*
PIN 2 = B ; /*
PIN 3 = S0 ; /*
PIN 4 = S1 ; /*
PIN 5 = R ; /*
PIN 6 = VAL ; /*
PIN 7 = EI ; /*ligar E0 a este input*/

/* ***** OUTPUT PINS ***** */
PIN 15 = OK ; /*
PIN 16 = E0 ;

P1 = A&B;
P2 = A#B;
P3 = A$B;

ResCer = (P1 & !S1 & !S0) # (P2 & !S1 & S0) # (P3 & S1 & !S0);
E0 = !(ResCer $ !R);

OK = !(EI & VAL) # (S0&S1);
```

Fig 3: Código CUPL

2.4 Resultado:

Finalmente, juntamos todas as partes do circuito:

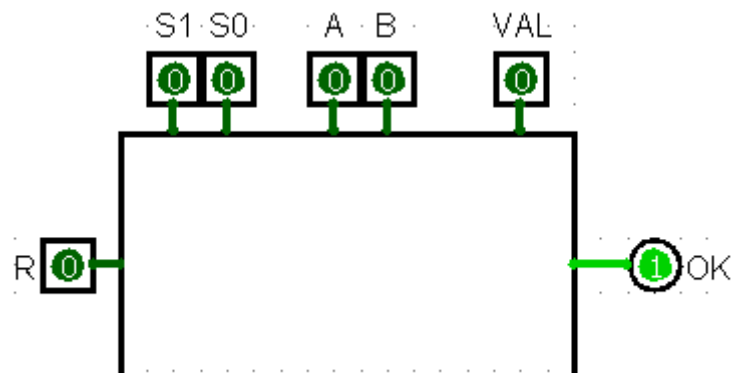


Figura 4: O sistema todo

3. Conclusões:

Em suma, todos os objetivos foram realizados. Os resultados finais foram consistentes com o que planeámos para a resolução do problema. Nós entendemos a lógica do exercício e apreciamos a sua realização.