Instituto Superior de Ciências do Trabalho e da Empresa

Departamento de Ciências e Tecnologias da Informação

Arquitectura de Computadores

Circuitos Combinatórios Típicos

(Descodificadores, Multiplexers, …)

Versão 0.01a

1. INTRODUÇÃO 2

2. DESCODIFICADOR 4

2.1. FUNCIONAMENTO INTERNO 4
2.2. CONSTRUIR UM DESCODIFICADOR A PARTIR DE OUTRO(S) 6
2.3. DESENHAR UM CIRCUITO COM BASE NUM DESCODIFICADOR 7

3. CODIFICADOR 9

3.1. FUNCIONAMENTO INTERNO 9

4. MULTIPLEXER 10

4.1. ESQUEMA INTERNO E FUNÇÃO LÓGICA 11
4.2. CONSTRUIR UM MULTIPLEXER A PARTIR DE OUTRO(S) 12
4.3. IMPLEMENTAR UMA FUNÇÃO COM UM MULTIPLEXER 12

5. DESMULTIPLEXER 14

6. COMPARADOR 14

7. ADICIONADORES E SUBTRACTORES 14

1. Introdução

Dizemos que um circuito é combinatório se, para cada combinação dos valores das entradas, as saídas são sempre as mesmas, independentemente do instante do tempo. Por outras palavras, poderemos dizer que o circuito não tem memória.

Por exemplo, considere um circuito com 2 entradas X e Y e duas saídas A e B. Considere também que ao experimentar o circuito com X=0 e Y=0 deu A=1 e B=0, então poderá garantir que sempre que as entradas forem X=0 e Y=0 então as saídas serão sempre A=1 e B=0.

Na verdade, se estivermos perante um circuito lógico combinatório, poderemos sempre fazer a sua tabela de verdade, relacionando assim as suas entradas com as suas saídas. Sendo assim, todas as portas lógicas e funções lógicas que foram anteriormente estudadas são circuitos combinatórios.

Exemplo: Imagine que tinha uma porta com um código introduzido por meio de 10 interruptores. Para verificar se o código estava correcto, bastaria um circuito para comparar cada um dos bits correspondentes a cada um dos interruptores. Este circuito corresponde um comparador de 10 bits, que é um tipo de circuito combinatório bem conhecido. A tabela de verdade deste circuito teria 10 variáveis, 1024 linhas e a função F só teria o valor lógico "1" numa dessas linhas.

Devido ao facto do seu uso ser muito frequente e geral, existe um conjunto de circuitos combinatórios bastante conhecidos e que podem ser usados, tais como usamos as portas lógicas, noutros circuitos.

Exemplo de circuitos combinatórios típicos:

- Comparadores
- Descodificadores e Codificadores
- Multiplexers de Desmultiplexers
- Adicionadores e Subtractores

Alguns destes circuitos serão abordados nas próximas secções, sob o ponto de vista da construção, funcionamento e utilização.

Na seguinte tabela é feita uma breve descrição de alguns destes circuitos.

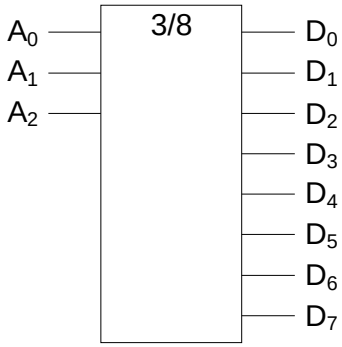
Circuito	Entradas	Saídas	Descrição
Descodificador	n	2^n	Introduzindo um número de n bits na entrada, activa a linha de saída correspondente. As restantes linhas ficam inactivas. **Exemplos**: desc 2/4, desc 3/8, desc 4/16, …
Codificador	2^n	n	O oposto ao descodificador. Activando uma única linha das suas entradas, obtém-se na saída o número binário correspondente a essa linha. Pode dizer-se que faz uma codificação, pois a saída será o valor numérico correspondente ao índice da linha activa. **Exemplos**: cod 4/2, cod 8/3, cod 16/4
Multiplexer	n + Linhas de selecção	1	Dadas n linhas de entrada, o multiplexer faz com que apenas uma delas fique ligada à saída, isto é, o valor da saída será o valor da entrada seleccionada. O multiplexer com n linhas de entrada, possui também *log(n)* linhas de selecção. Por exemplo, o multiplexer 8-1 tem 8 linhas de entrada + 3 linhas de selecção. Note que log(8) = 3 **Exemplos**: Mux 4-1; Mux 8-1; Mux 16-1; …
Desmultiplexer	1 + Linhas de controlo	n	O circuito oposto ao multiplexer. Tem uma única entrada, n saídas e log(n) linhas de controlo. O circuito permite direccionar a sua única entrada para a linha indicada pelo selector. **Exemplos**: Dx 1-4; Dx 1-8; Dx 1-16
Comparador	$n + n$	1	Permite comparar 2 números de n bits. O circuito tem uma única saída que será activa caso os dois números sejam iguais e desactiva caso contrário. **Exemplo**: Comparador 4 bits - recebe dois números de 4 bits
Adicionador	$n + n + c_{in}$	$n + c_{out}$	Permite fazer a soma de dois números de n bits. **Exemplo**: Adicionador de 4 bits – faz a soma de dois números de 4 bits

2. Descodificador

Um descodificador é um circuito combinatório com *n* entradas e 2^n saídas. Ao introduzir um número à entrada, o descodificador activa a linha de saída correspondente a esse número. As restantes linhas ficam inactivas. Apenas uma saída virá com valor lógico diferente de todas as outras. Um descodificador com *n* entradas e 2^n saídas diz-se um *descodificador n para* 2^n (2-para-4, 3-para-8, etc.)

A ideia é a seguinte: se introduzirmos o valor *x* (em binário) na entrada, a saída número *x* será activada, sendo que as saídas estão numeradas de 0 a *n*-1.

A figura seguinte apresenta um descodificador 3-para-8, pois tem 3 entradas e 8 saídas ($2^3 = 8$).



Exemplo: Considerando a figura anterior.

> O que acontecerá se o valor da entrada for 001 (A_2=0, A_1=0, A_0=1)? - A única saída que ficaria activa seria a D_1, as restantes saídas ficariam todas com o valor lógico "0".
>
> Considere agora que a entrada é 101. Note que $(101)_2 = (5)_{10}$, logo todas as saídas ficariam com o valor lógico "0", excepto D_5 que teria o valor lógico "1".

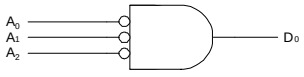
Existem outros circuitos combinatórios que habitualmente se designam por descodificadores, como é o caso do descodificador BCD - 7 segmentos e do Descodificador BCD - Excesso-3. Estes circuitos não são na verdade descodificadores como anteriormente definimos. Na realidade seria mais correcto designá-los por *conversores de código*.

2.1. Funcionamento interno

Considere que pretende fazer o descodificador 3-8 apresentado na figura anterior.

Qual o circuito que utilizaria para produzir a saída D_0?

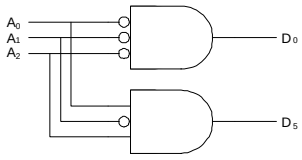
- Como sabe a saída D_0 só é activa quando a entrada é 000 (A_2=0, A_1=0, A_0=0).
- Assim, $D_0 = \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0}$ (repare que na tabela de verdade só a primeira linha seria "1")
- Ou seja, o circuito seguinte resolveria o problema



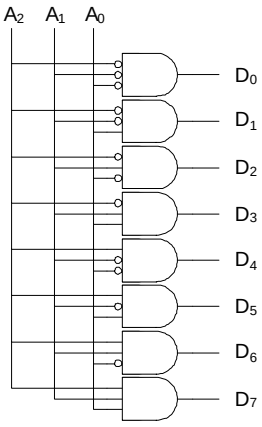
e para produzir a saída D_5?

– A saída D_5 é activa quando se introduz o número 5 à entrada, ou seja com a combinação 101.
– Resulta que $D_5 = A_2 \cdot \overline{A_1} \cdot A_0$ (repare que na tabela de verdade só a sexta linha seria "1")

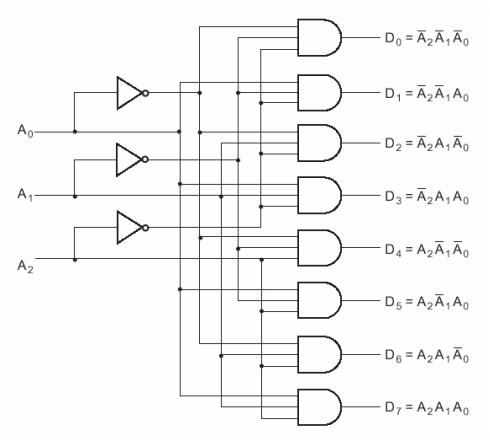
o circuito seguinte resolveria o problema para as saídas D_0 e D_5.



Seguindo este raciocínio chegaríamos aos circuitos das figuras seguintes, que representam o esquema interno de um descodificador 3-8.



ou

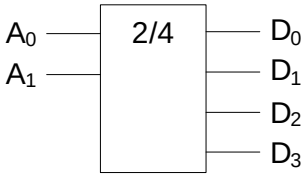


Note que os dois circuitos anteriores são equivalentes, mas o segundo usa apenas três inversores enquanto o primeiro usa três para o primeiro AND, dois para o segundo, etc.

A tabela de verdade (que relaciona as entradas com as saídas) do descodificador 3-8 é a seguinte:

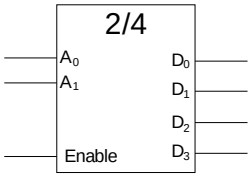
A_2	A_1	A_0	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	**1**	0	0	0	0	0	0	0
0	0	1	0	**1**	0	0	0	0	0	0
0	1	0	0	0	**1**	0	0	0	0	0
0	1	1	0	0	0	**1**	0	0	0	0
1	0	0	0	0	0	0	**1**	0	0	0
1	0	1	0	0	0	0	0	**1**	0	0
1	1	0	0	0	0	0	0	0	**1**	0
1	1	1	0	0	0	0	0	0	0	**1**

Exercício: Faça a tabela de verdade de um descodificador 2-4.



2.2. Construir um descodificador a partir de outro(s)

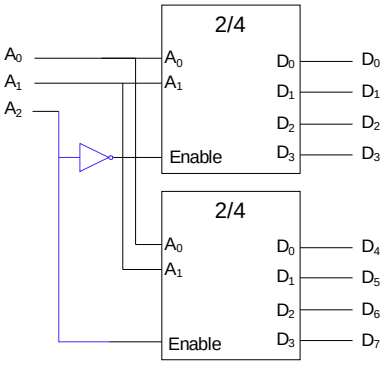
Considere que se pretende utilizar um descodificador 3/8, mas apenas estão disponíveis vários descodificadores 2/4 com uma entrada *enable*. Como resolveria o seu problema?



Repare que:

- Quando o bit mais significativo (A_2) é "0", as quatro últimas saídas ficam desactivadas.
- Quando o bit mais significativo é "1", as quatro primeiras saídas ficam desactivadas.

Assim, o bit mais significativo poderá usado para seleccionar as quatro primeiras ou últimas saídas. O circuito seguinte mostra o resultado pretendido.



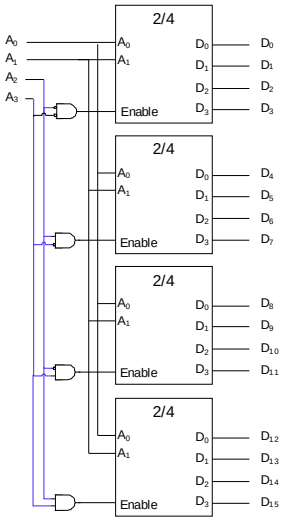
Exercício: Como se faria um descodificador 4-16 a partir de vários 2-4?

Basta pensar que os 2 bits mais significativos seleccionam o 1º, 2º, 3º ou 4º grupo de 4 linhas da tabela de verdade.

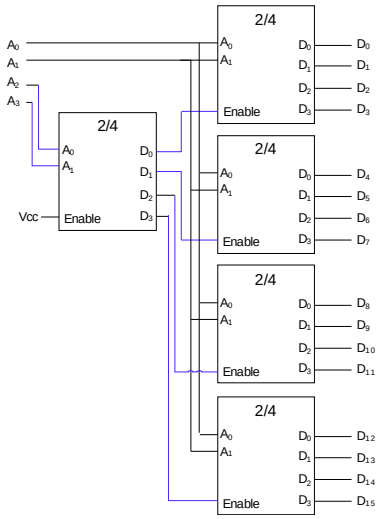
A_3	A_2	Linhas activas
0	0	1º grupo de 4 linhas
0	1	2º grupo de 4 linhas
1	0	3º grupo de 4 linhas
1	1	4º grupo de 4 linhas

Assim, os dois bits mais significativos estariam associados às entradas de *enable*, enquanto os menos significativos seriam todos ligados em paralelo às duas entradas de cada um dos descodificadores.

Note também que o circuito para seleccionar cada um dos descodificadores pode ser visto como um descodificador de 2-4.



ou

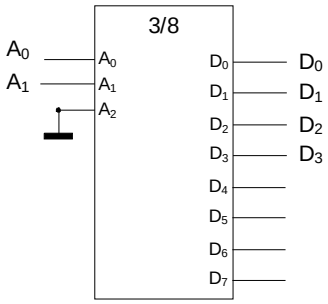


Exercício: Como construiria um descodificador 2/4, com base num 3/8?

Note que todos os valores das saídas para a segunda metade da tabela da verdade são '0'.

Assim, uma vez que só nos interessam as saídas para as quais o bit mais significativo é '0', basta ligar o bit mais significativo a '0' e aproveitar apenas a primeira metade das saídas.

Outra alternativa seria ligar o bit mais significativo a '1' e aproveitar a segunda parte das saídas.



2.3. Desenhar um circuito com base num descodificador

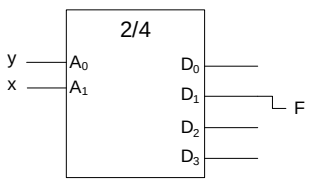
Qualquer função lógica pode ser realizada utilizando apenas um descodificador e portas OR.

Exemplo: Considere a função $F = \overline{x}.y$

A tabela de verdade de F tem apenas um termo com valor lógico '1'.

Como falamos de uma função de duas variáveis, utilizaremos um descodificador 2-4.

A_3	A_2	F
0	0	0
0	1	1
1	0	0
1	1	0



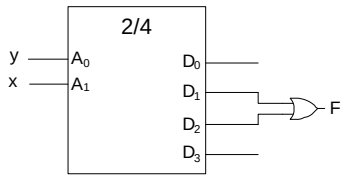
Qual é o valor de D_1 quando a entrada é 00? – é '0' pois a única saída activa é D_0.

Qual é o valor de D_1 quando a entrada é 01, 10 e 11? Verifique que o circuito obedece à Tabela.

Considere agora que se pretende fazer um circuito que realize a função $F = \bar{x}.y + x.\bar{y}$.

Neste caso a função será '1' em duas situações distintas: quando xy é 01 e 10. Assim, a função F será a soma de D_1 e D_2. A figura seguinte apresenta o circuito resultante.

A_3	A_2	F
0	0	0
0	1	1
1	0	1
1	1	0



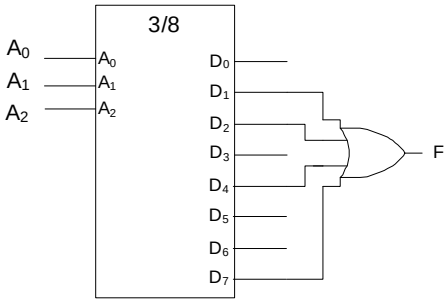
Consideremos agora uma função qualquer de *n* variáveis. Uma vez que as saídas do descodificador correspondem aos termos mínimos de uma tabela de verdade, para implementar uma função de *n* variáveis, utilizando um descodificador, basta juntar as saídas correspondentes aos termos mínimos com recurso a portas OR.

Exemplo: Pretende-se fazer um circuito que indique se a soma dos 3 bits de entrada é ímpar.

A função pretendida obedece à tabela de verdade seguinte.

O circuito correspondente usando um descodificador poderia ser:

A_2	A_1	A_0	F
0	0	0	**0**
0	0	1	**1**
0	1	0	**1**
0	1	1	**0**
1	0	0	**1**
1	0	1	**0**
1	1	0	**0**
1	1	1	**1**



Os descodificadores são também adequados para implementar circuitos com várias saídas. Basta para isso adicionar portas OR.

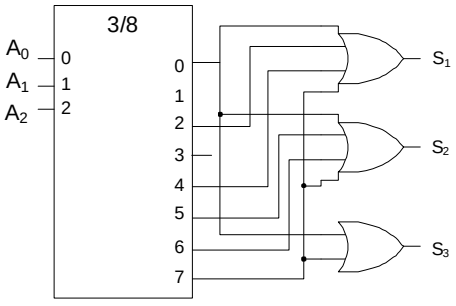
Exemplo: Suponha que se pretende realizar um circuito para um elevador com três saídas:

A primeira indica se o andar em que estamos é ímpar.

A segunda indica se o andar é o R/C ou um dos 3 pisos superiores: 0, 5, 6, 7

A terceira indica se o elevador só pode andar numa direcção (1º e último andar)

A_2	A_1	A_0	S_1	S_2	S_3
0	0	0	**0**	**1**	**1**
0	0	1	**1**	**0**	**0**
0	1	0	**1**	**0**	**0**
0	1	1	**0**	**0**	**0**
1	0	0	**1**	**0**	**0**
1	0	1	**0**	**1**	**0**
1	1	0	**0**	**1**	**0**
1	1	1	**1**	**1**	**1**

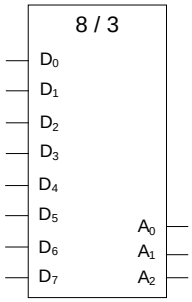


3. Codificador

O codificador é um circuito que realiza a operação inversa do descodificador, tem 2^n entradas e *n* saídas. Quando se trabalha com codificadores parte-se do princípio que apenas uma das entradas está activa num dado instante de tempo. A saída corresponde ao número da entrada que está activa. Por exemplo, se activar a linha 5 o resultado será o número $(101)_2 = 5$.

Exemplo: Codificador 8-para-3, juntamente com a respectiva tabela de verdade.

Note que a tabela de verdade não está completa. Para os casos que faltam as saídas são '0'.



D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	A_2	A_1	A_0
1	0	0	0	0	0	0	0	0	0	0
0	**1**	0	0	0	0	0	0	0	0	1
0	0	**1**	0	0	0	0	0	0	1	0
0	0	0	**1**	0	0	0	0	0	1	1
0	0	0	0	**1**	0	0	0	1	0	0
0	0	0	0	0	**1**	0	0	1	0	1
0	0	0	0	0	0	**1**	0	1	1	0
0	0	0	0	0	0	0	**1**	1	1	1

3.1. Funcionamento interno

O codificador pode ser construído usando apenas portas OR.

Repare, por exemplo, que o bit menos significativo da saída (A_0) fica activo quando uma das entradas D_1, D_3, D_5, D_7 está activa. Note também que está apenas uma entrada activa de cada vez.

Assim $A_0 = D_1 + D_3 + D_5 + D_7$

Exercício: Suponha que pretende construir um codificador 8-3 como o da figura anterior.

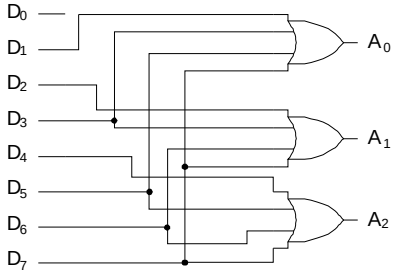
As expressões para as saídas vêm directamente da observação da tabela de verdade

$A_0 = D_1 + D_3 + D_5 + D_7$

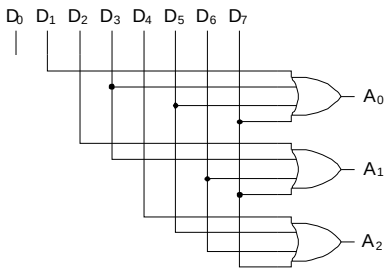
$A_1 = D_2 + D_3 + D_6 + D_7$

$A_2 = D_4 + D_5 + D_6 + D_7$

O circuito resultante é apresentado na figura seguinte.



ou



Exercício: Considere que tem apenas um codificador 8-para-3.

Como faria para o utilizar como um codificador 4-para-2?

4. Multiplexer

O multiplexer é um circuito combinatório cujo propósito é ligar a sua única saída a uma das *n* entradas possíveis. Por outras palavras, selecciona uma das *n* entradas e liga-a à saída.

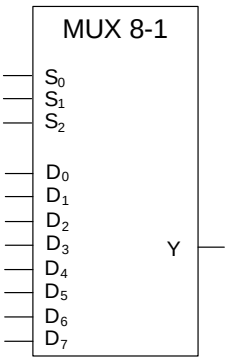
A selecção é feita através de um conjunto de selectores, ou seja, linhas de controlo.

Um multiplexer de *n* entradas tem log(n) linhas de selecção. Por outras palavras, um multiplexer com *m* linhas de selecção, tem 2^m linhas de entrada.

Porque é que um multiplexer tem *m* linhas de selecção para 2^m linhas de entrada?

Basta pensar por exemplo para no caso de 8 linhas de entrada. – Quantos bits são necessários para contar de de 0 até 7? O princípio é este: se temos *n* linhas de entrada, o número de linhas no selector será o número mínimo de bits necessário para escrever qualquer número de 0 a *n*-1.

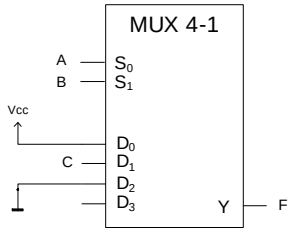
Exemplo: O multiplexer 8-para-1 tem 8 entradas e 3 linhas de selecção. ($2^3 = 8$ ou seja, log(8) = 3).



Exercício: Considere o multiplexer de 8-para-1 da figura anterior.

Qual é o valor da saída se o selector S= $S_2S_1S_0$ tiver o valor 000? - Será o valor que estiver em D_0
Qual é o valor da saída se o selector S= $S_2S_1S_0$ tiver o valor 101? - Será o valor que estiver em D_5

Exercício: Considere o multiplexer 4-1 da figura seguinte:



- Qual é o valor de *F* se A=0, B=0, C=0?
  Como A=0 e B=0, a linha D_0 fica seleccionada. Como D_0 está ligada a Vcc, sai '1'.
  Neste caso o valor de C não interessa pois as linhas D_1, D_2 e D_3 estão desactivadas.
- Qual o valor de F quando A=1 e B=0?
  Como B corresponde ao bit mais significativo, o número introduzido no selector é 01, assim a entrada seleccionada será D_1, como o valor que entra em D_1 é C, F=C
- Qual o valor de F quando o selector é 11?
  Neste caso será um valor indefinido, pois nada está ligado à entrada D_3.

4.1. Esquema interno e função lógica

Suponha que pretende fazer um multiplexer, por exemplo 4-para-1. À partida poderemos logo dizer que o nosso circuito terá 4 linhas de entrada + 2 linhas de selecção e uma saída.

Quando as linhas de selecção forem 00, pretende-se que F = D_0, quando forem 01, F será D_1 …

S_1	S_0	Y
0	0	**D_0**
0	1	**D_1**
1	0	**D_2**
1	1	**D_3**

Na tabela acima pode-se verificar por exemplo que, quando o selector for 00 então Y=D_0. Ora, isto quer simplesmente dizer que se D_0 = 0 então Y=0, se D_0 = 1 então Y=1,e por aí adiante. Seguindo este raciocínio, a tabela anterior pode escrever-se na forma:

S_1	S_0	D_0	D_1	D_2	D_3	F
0	0	0	x	x	x	**0**
0	0	1	x	x	x	**1**
0	1	x	0	x	x	**0**
0	1	x	1	x	x	**1**
1	0	x	x	0	x	**0**
1	0	x	x	1	x	**1**
1	1	x	x	x	0	**0**
1	1	x	x	x	1	**1**

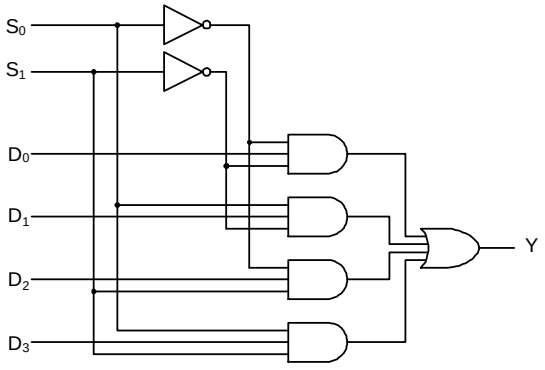
Assim, Y será '1' se:

- As linhas de selecção forem 00 e D_0 = 1, ou
- As linhas de selecção forem 01 e D_1 = 1, ou
- As linhas de selecção forem 10 e D_2 = 1, ou
- As linhas de selecção forem 11 e D_3 = 1

Concluindo, a função lógica para um multiplexer 4-1 vem:

$$Y = \overline{S_1}\,\overline{S_0}D_0 + \overline{S_1}S_0D_1 + S_1\overline{S_0}D_2 + S_1S_0D_3$$

O circuito está representado na figura seguinte, e corresponde ao esquema interno de um MUX 4-1.

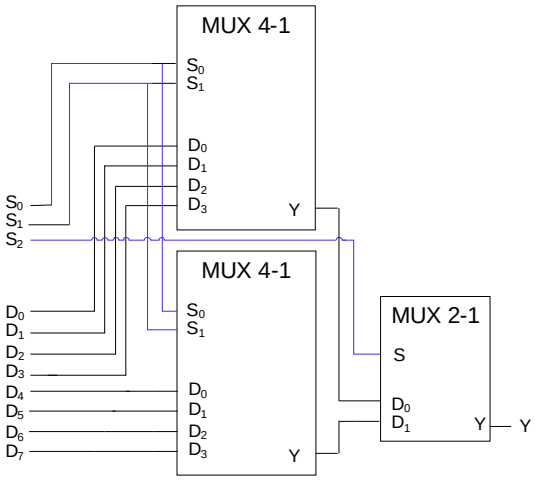


4.2. Construir um multiplexer a partir de outro(s)

À semelhança do que se disse sobre os descodificadores, tendo por base multiplexers de um certo tipo, é também possível construir multiplexers de outro tipo.

Exemplo: Suponha que tem vários multiplexers 4-1, mas necessita de um multiplexer 8-3.

Uma solução possível pode ser a que se apresenta na figura seguinte. Neste caso o bit mais significativo do selector é ligado a um multiplexer 2-1 que o que faz é seleccionar o multiplexer que efectivamente se está a utilizar. Cada um dos MUXs 4-1 faz metade da tabela da verdade, o MUX 2-1 apenas selecciona qual a metade que se pretende.



(EXPLICAR)

4.3. Implementar uma função com um multiplexer

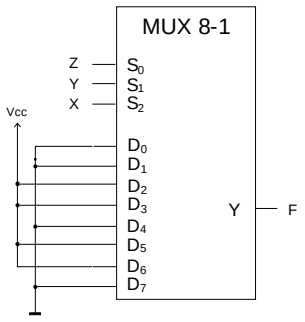
Qualquer função lógica de n variáveis pode ser facilmente implementada com recurso a um multiplexer com *n* selectores, isto é, um multiplexer 2^n-para-1.

Exemplo: Considere a função $F(X,Y,Z) = \sum m(2,3,5,6)$.

A função tem 3 variáveis como tal, usaremos um MUX 8-1 para a implementar.

A tabela de verdade seguinte corresponde à função F. Ao lado encontra-se o circuito implementado com um multiplexer. Note que X será o bit mais significativo do selector do multiplexer.

X	Y	Z	F
0	0	0	**0**
0	0	1	**0**
0	1	0	**1**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**0**



Pode-se verificar facilmente que o circuito apresentado satisfaz a função F. Tomem-se, por exemplo, os valores X=1, Y=1, Z=0. Sendo X o bit mais significativo temos $(110)_2$=6. Com os selectores assim definidos, a saída do multiplexer será o que estiver em D_6. Pela tabela de verdade pode ver-se que o valor de F para essa linha deverá ser 1. Assim, basta forçar que a entrada D_6 do multiplexer seja 1 (como ? – ligando-a a Vcc).

É também possível implementar com facilidade qualquer função lógica de *n* variáveis, recorrendo a um multiplexer com *n-1* selectores, isto é, um multiplexer 2^{n-1}-para-1. Neste caso, as entradas do multiplexer poderão passar a depender da n-ésima variável que não foi usada nas linhas do selector.

Exemplo: Considere a função $F(X,Y,Z) = \sum m(2,3,5,6)$ do exercício anterior.

X	Y	Z	F
0	0	0	**0**
0	0	1	**0**
0	1	0	**1**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**0**

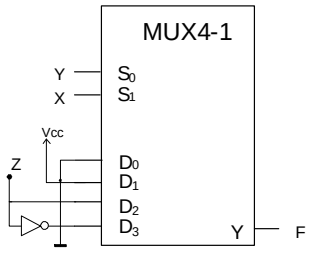
Vamos implementar essa função, mas agora usando um MUX 4-1.

A estratégia será usar duas das variáveis nas linhas dos selectores e fazer com que as entradas dependam da terceira. Por exemplo, podemos considerar apenas X e Y para as entradas dos selectores.

Tentaremos fazer uma tabela de verdade com duas variáveis, equivalente à anterior

- Qual o valor de F quando X=0 e Y=0? – é sempre '0'
- Qual o valor de F quando X=0 e Y=1? – é sempre '1'
- Qual o valor de F quando X=1 e Y=0?
  Depende do valor de Z. Quando Z=0 vem F=0. Quando Z=1 vem F=1. Assim F=Z
- Qual o valor de F quando X=0 e Y=1?
  Depende do valor de Z. Quando Z=0 vem F=1. Quando Z=1 vem F=0. Assim $F = \overline{Z}$

X	Y	F
0	0	**0**
0	1	**1**
1	0	**Z**
1	1	$\overline{Z}$

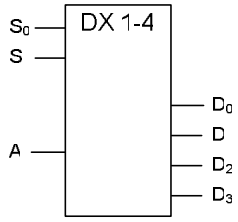


5. Desmultiplexer

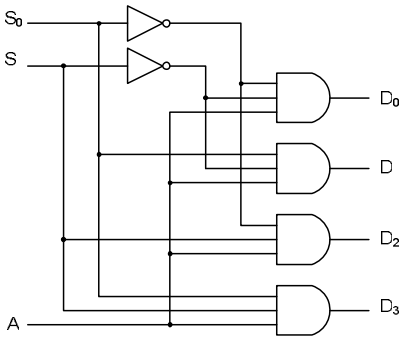
Circuito que realiza a operação inversa de um multiplexer.

Direcciona a entrada para uma de 2^n linhas de saída, que é seleccionada através de *n* linhas de controlo

Exemplo: Desmultiplexer 1-para-4 linhas



Esquema interno



6. Comparador

Um comparador serve para comparar *n* bits, têm *n* + *n* entradas e uma saída.

(POR FAZER)

7. Adicionador e Subtractor

(POR FAZER)

Registo de Revisões

Ver 0.01 - 26-01-2005 – Fernando Batista
Versão draft inicial
Publicação antecipada para o período de avaliações 2004/2005
Ver 0.01a - 27-01-2005 – Tomás Brandão
Versão draft inicial com algumas gralhas corrigidas