Mestrado Integrado em Engenharia Biomédica

Microtecnologias no Silício

Trabalho Prático

Ana Luisa Oliveira
16-01-2015

Introdução

Este trabalho consiste numa estrutura *Master Slave* que a permitir comunicação de vários sensores com o *master*. A comunicação é feita em código *Manchester* e utiliza duas linhas em *open-drain* (linha de relógio e linha de dados). O *master* é quem determina qual o sensor que vai colocar os dados na linha e quando é que a comunicação se inicia ou se deixa de fazer.

A codificação *Manchester* (ver tabela 1) permite o sincronismo entre o *busdata* e o *clock* uma vez que há sempre mudança de nivel a cada ciclo de relógio. Assim não é necessario enviar bits para controlo de erros uma vez que estes podem ser facilmente detetados pela falta de sincronismo. Este tipo de codificação é vantojoso porque, por exemplo, se o código enviado fosse o nivel lógico "0" e "1" correspondente a 0 e 5 V, no caso de existir falha na comunicação, o *receiver* iria interpretar o que estava a receber como "0" lógico, o que não deveria acontecer. Isto não acontece na codificação *Manchester*.

Tabela 1 Codificação Manchester

Clk	10	1
Dados	10	
Clk	10	0
Dados	01	
Clk	10	Idle
Dados	11	
Clk	10	Nulo
Dados	00	

As linhas em *open-drain* têm resistências *pull-up* de modo à linha ter o nivel logico "1". Deste modo possibilita-se que vários sensores estejam ligados à mesma linha e que possam ler ou escrever nela.

Todos os sensores vão estar a ler a informação da linha e quando reconhecem que são chamados pelo *master*, enviam um bit de reconhecimento e procedem ao envio dos dados. Assim a linha de dados tem de ser bidirecional e garantir que os dados que estao a ser escritos na linha sejam provenientes de um só sensor, portanto, tem de ser em *open-drain*. No entanto a linha de *clock* não tem de o ser obrigatoriamente uma vez que os sensores só vão ler a partir dela.

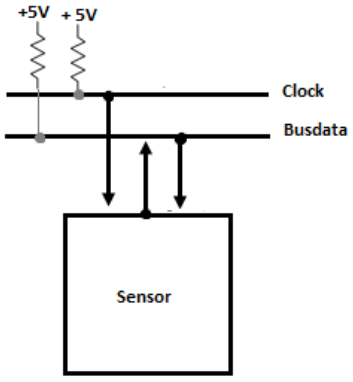

Figura 1 Esquema da comunicação

A linha de dados inicialmente encontra-se a “11” (em idle) e quando o *master* pretende iniciar a comunicação envia o *start bit* que na linha corresponde a “01” (0 em *Manchester*). Os sensores ficam à espera dos dois bits seguintes que correspondem aos bits de endereçamento. Se estes bits corresponderem ao seu endereço, o sensor envia o sinal de reconhecimento “10” (*acknowlege*) a informar o *master* que reconheceu o seu chamamento e procede ao envio dos dados. Esta estrutura de comunicação encontra-se visivel no esquema abaixo.

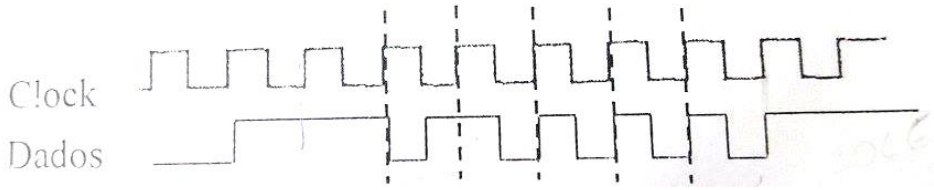

Figura 2 Estrutura de dados

Neste trabalho, foi realizada a estrutura para um só sensor, com o endereço 11 e que envia “1’s” lógicos. No entanto, como se tem 2 bits de endereço, pode-se ter até 4 endereços diferentes e portanto 4 sensores.

A estrutura foi criada utilizando o programa *S-Edit*. Como entrada no sistema para a simulação foi colocado o sinal “00111101101010101111” (figura 5) numa fonte de sinal com as configurações visíveis na figura 3. O clock (figura 5) foi gerado com repetições da sequência “10” e com as configurações da figura 4. Os parametros FT e RT são colocados com valores muito reduzidos de forma a que as transiçoes de nivel lógico dos sinais se deêm o mais rápido possível.

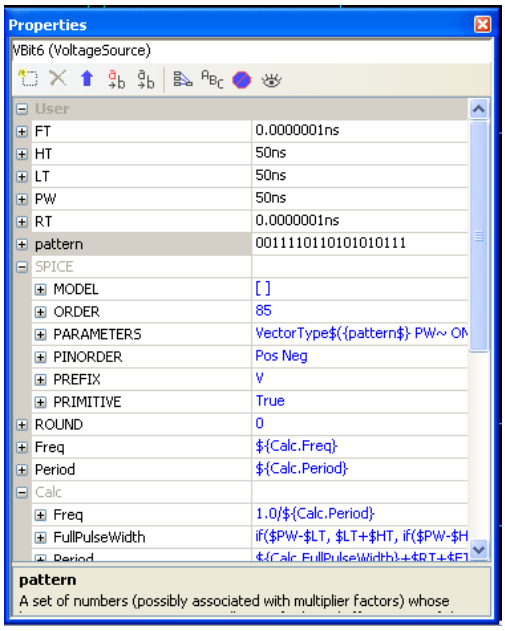

Figura 4 Configurações do sinal busdata

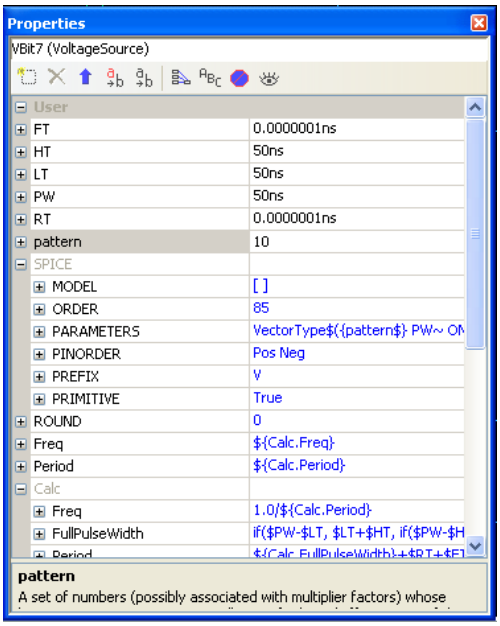

Figura 3 Configurações do sinal clock

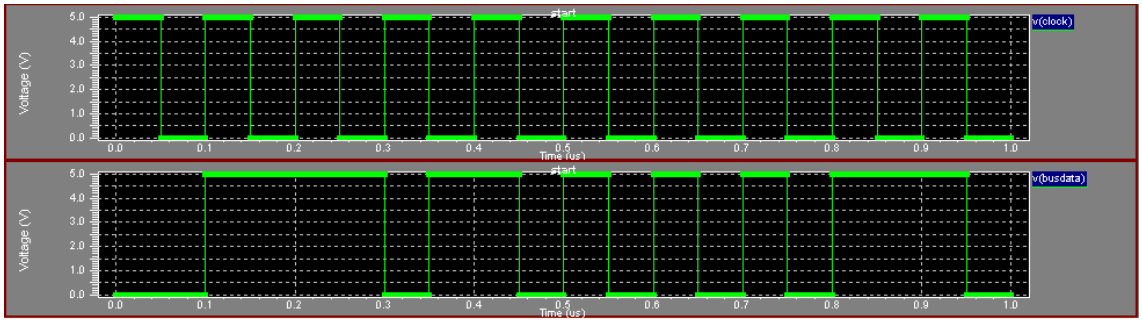

Figura 5 Sinal clock e busdata

Descodificador Manchester

Um primeiro componente necessario é um descodificador *Manchester* para converter os dados que são recebidos.

O *busdata* entra numa *p-latch* que é obtida colocando um inversor antes da entrada do sinal do clock de uma *n-lacth*. Usa-se uma *p-latch* porque, ao contrário dos *flip-flops*, elas respondem durante todo o ciclo e não apenas durante a subida ou descida. Para alem disso, têm memória, o que é imprescindivel na descodificação *Manchester*.

A *p-lacth* permite a passagem do que está na entrada quando o nivel lógico do *clock* é "1" e envia o estado anterior quando o nivel lógico do *clock* é "0", ignorando a entrada. Assim à saida de *p-lacth* tem-se "00" se a entrada for um nulo ou o nível lógico "0" e "11" se a entrada for um *idle* ou o valor lógico "1". Com isto consegue-se adescodificação do primeiro bit.

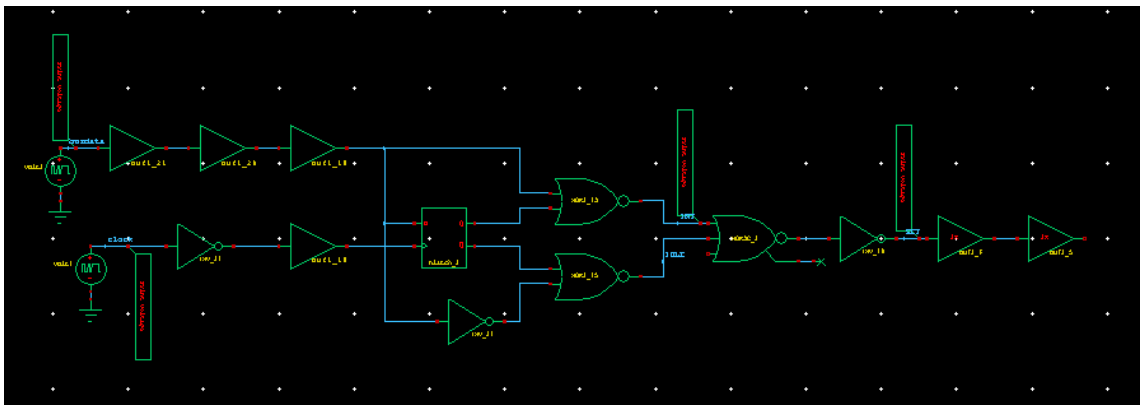

Figura 6 Descodificador Manchester no S-Edit

Nas portas lógicas seguintes, as saidas Q e Q negada da *latch* são comparadas com o sinal de entrada de forma a se distinguir entre valor lógico e nulo ou idle. Sai "10" se o sinal que está a entrar for valor lógico ou "11" caso contrário.

Na primeira NOR a saida Q da *latch* é comparada com o *busdata* de forma a se distinguir entre o nulo e o valor lógico zero. Assim o sinal INT vai ser "11", "10" ou "00" se a entrada for nulo, "0" lógico ou um dos restantes casos, respetivamente.

Na segunda NOR a saida Q negada é comparada com o *busdata* de forma a se distinguir entre o *idle* e o valor lógico "1". Assim o sinal IDLE vai ser "11" se a entrada for *idle*, "10" se for o valor lógico "1" ou "00" se for uma das restantes hipóteses.

A entrada dos sinais INT e IDLE numa OR resulta na geração do sinal SET que toma o valor "10" em caso de valor lógico e "11" em caso contrário. Este sinal irá ser aplicado no controlo dos *flip-flops*. Esta OR posteriomente irá receber também o sinal ERROR de forma a que, no caso de existir um erro (por exemplo, perda do sincronismo entre os vários sistemas), o sinal SET seja ativado e assim haja o *set* dos *flip-flops*.

Tabela 2 Resposta da p-latch, das NOR e da OR

Clk	Busdata	Q	Q negada	INT	IDLE	SET
10	**00**	00	11	11	00	11
10	**01**	00	11	10	00	10
10	**10**	11	00	00	10	10
10	**11**	11	00	00	11	11

São colocados *buffers* na saída do *clock*, do *busdata* e do SET para que os sinais tenham potência suficiente para serem aplicados noutros pontos do sistema.

Os sinais IDLE, INT e SET obtidos no fim do descodificador *Manchester* são visíveis nos gráficos seguintes.

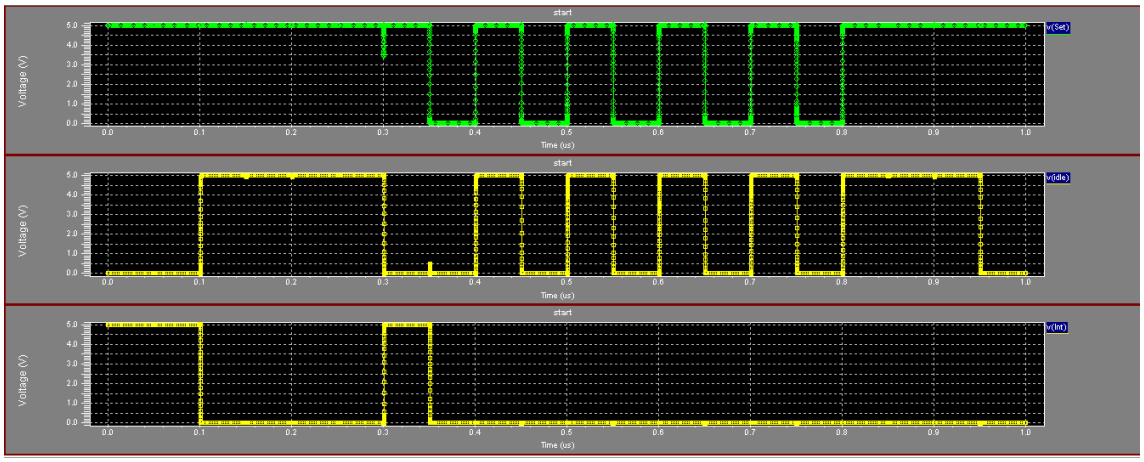

Figura 7 Sinais SET, IDLE e INT

Startbit e Endereçamento

O circuito que permite a discriminação do *startbit* e do endereçamento enviado pelo master encontra-se na figura 8.

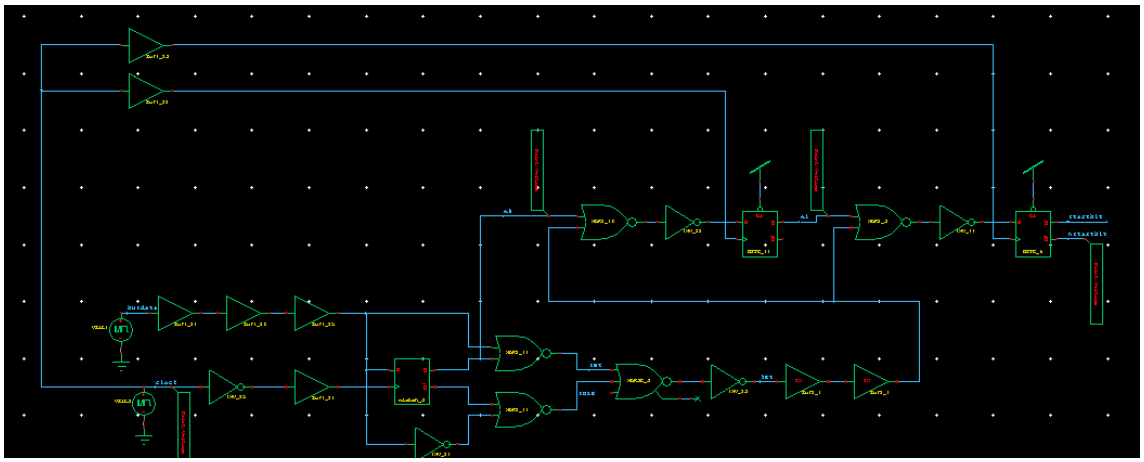

Figura 8 Circuito que permite a recolha dos bits de endereçamento e do startbit

Quando o sinal SET tem o valor "11" é feito o *set* dos *flip-flops* e estes vão ter "1" como saida, ou seja, a linha é mantida com o valor lógico "1". Quando o SET tem valor "10", os *flip-flops* vão ter como saída "1" ou "0" de acordo com o valor lógico de entrada.

Quando se dá o envio do startbit "01" , a saida da *latch* é "00" e o valor tomado pelo SET é "10", assim o valor resultante da porta lógica OR será "10" e portanto a saida do *flip-flop* é "10". Ou seja, a linha, que se encontrava a 1, é colocada a 0.

Tabela 3 Saida dos flip-flops

Clk	Busdata	Q	SET	*Flip-flop*
10	**00**	00	11	11
10	**01**	00	10	10
10	**10**	11	10	11
10	**11**	11	11	11

A seguir ao startbit aparecem os bits de endereçamento. Como os *flip-flops* causam desfasamento nos sinais e são usados 2 *flip-flops*, é posssivel ver o sinal *startbit* e os sinais com os bit de endereçamento (A0 e A1). Na saida do segundo *flip-flop* é visivel o sinal de *startbit*, à saida do primeiro vê-se o primeiro bit de endereçamento e na saida da *p-latch* vê-se o segundo bit de endereçamento.

O *startbit* indica o início da transmissão de dados e os bits de endereçamento permitem a identificação do sensor para o qual se pretende enviar a informação. O sensor ao qual o endereço corresponde terá de o reconhecer e enviar um sinal a indicar que o reconhecimento foi concretizado.

Os sinais obtidos podem ser visualizados na figura 9.

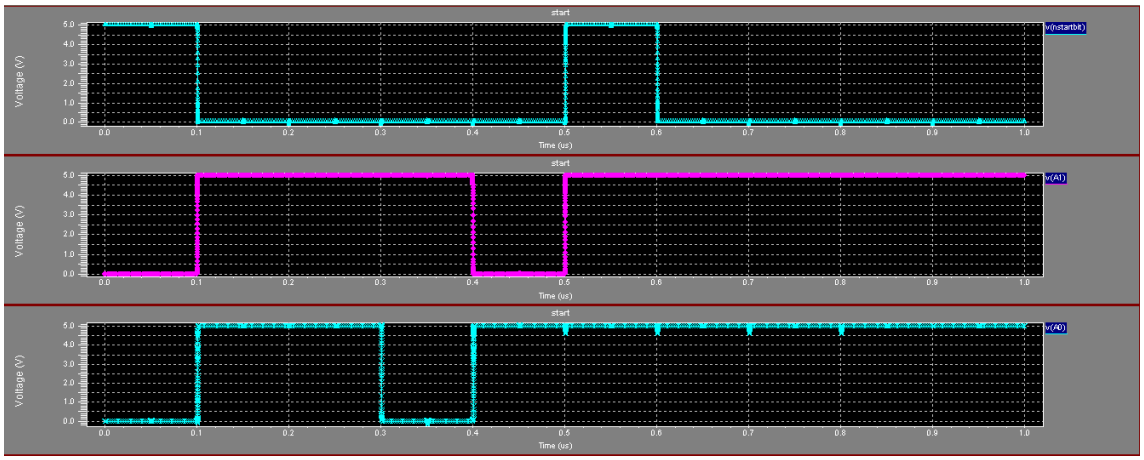

Figura 9 Sinal Startbit, A1 e A0

Sinal STOP

O circuito apresentado abaixo permite a obtenção do sinal STOP. Este sinal diferencia se existe ou não o *startbit*. Quando chegam dois *idles* o sinal altera-se para 1 e quando aparece um *startbit* altera-se para 0. É colocado um *flip-flop* para que o sinal seja atrasado um período de forma a que a alteração do sinal STOP só ocorra depois do *startbit*. Quando aparece um novo *idle* o sinal volta a mudar para 1.

O sinal obtido pode ser visualizado na figura 10.

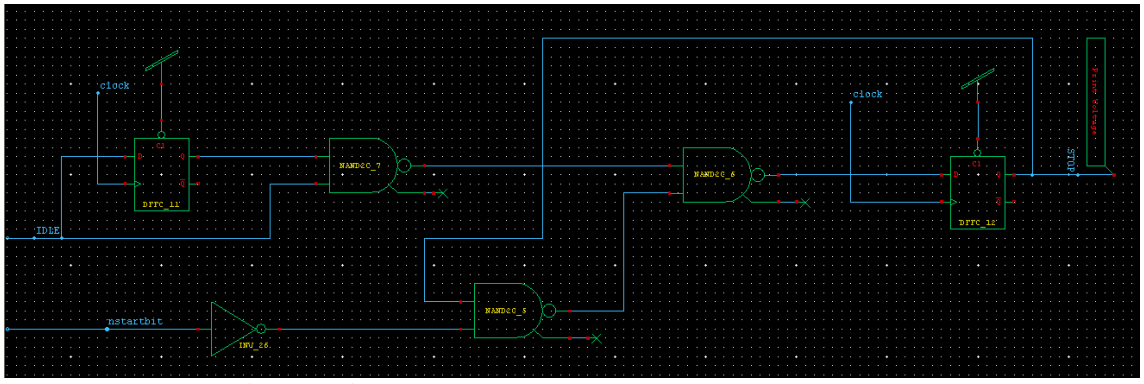

Figura 10 Circuito para obter o sinal STOP

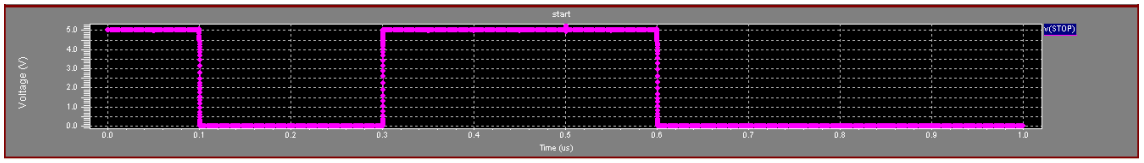

Figura 11 Sinal STOP e sinais que entram no circuito

Reconhecimento do endereço (Aknowledge)

O circuito apresentado dá o exemplo de um sensor cujo endereço é “11”. Este endereço é dado pelos *buffers*. Os sinais A0 e A1 são comparados com o endereço do sensor e resultam num sinal com o nível logico “1” à saída da XOR se forem iguais ao endereço. Se o endereço presente no *busdata* corresponder ao do sensor, se o sinal STOP estiver a “1” e o *startbit* também, o resultado à saída da NAND3 é “1” e é enviado o sinal *aknowledge* (ACK) (visível na figura 12). Este sinal é colocado no *busdata* e dá a indicação de que o sensor reconheceu que estava a ser chamado e que vai dar inicio ao envio dos dados.

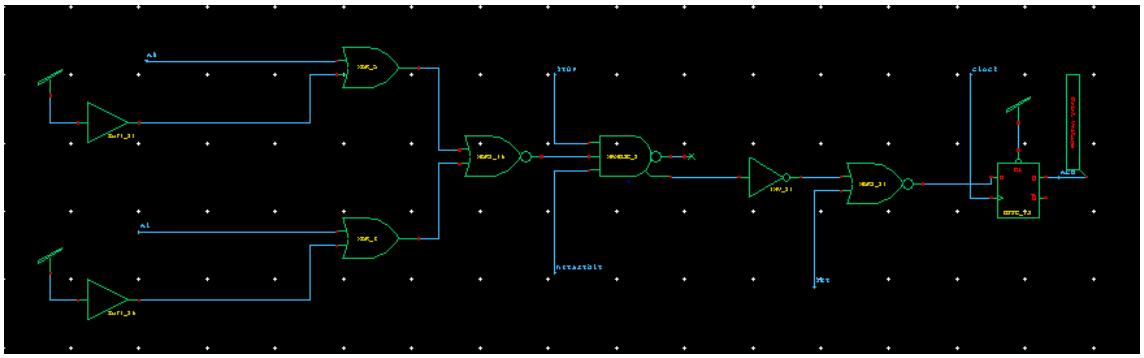

Figura 12 Circuito para obter o sinal ACK

É colocado um flip-flop com set para atrasar o sinal de forma a que o ACK apareça depois do *startbit*. O sinal set neste momento é “10” uma vez que corresponde à entrada do segundo bit de endereçamento e portanto, a um valor lógico. Este *flip-flop* é conseguido colocando um inversor e uma OR na entrada.

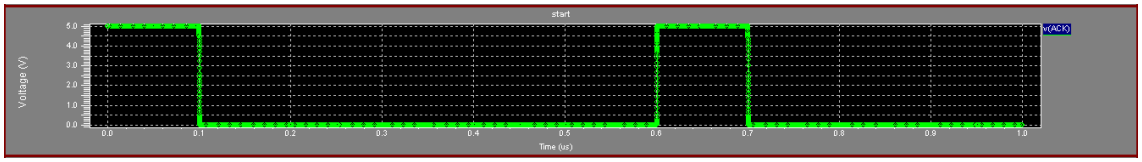

Figura 13 Sinal ACK

Sinal ENABLE

O sinal ENABLE permite ao sensor escrever na linha. Quando está a “1”, o sensor consegue enviar os dados, quando está a “0”, o sensor deixa de conseguir colocar os dados na linha de dados.

Inicialmente tanto o ACK como o ENABLE estão a zero. Quando o ACK passa a “1”, o ENABLE é ativado passando a “1”. Como se pretende que o ENABLE só apareça depois do ACK, é colocado um *flip-flop* para o atrasar.

O sinal ENABLE volta a “0” quando o set toma o valor “11”, ou seja, quando surge um *idle* no *busdata*, o que significa que a comunicação terminou e portanto o sensor deve deixar de enviar informação.

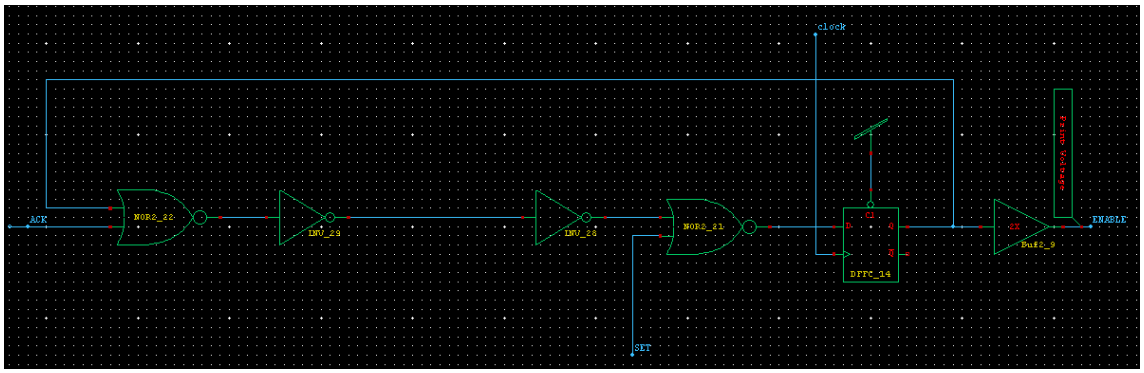

Figura 14 Circuito para a geração do sinal ENABLE

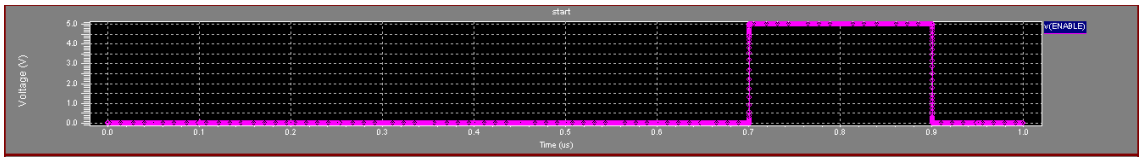

Figura 15 Sinal ENABLE

Codificador Manchester

Para que o sensor possa enviar os dados para a linha é necessário codificar primeiro os dados para código Manchester.

A codificação é feita com recurso a uma XNOR que tem como entradas o sinal *clock* e os dados enviados pelo sensor. Como este sensor está a enviar “1’s” lógicos e numa XNOR o resultado é 1 quando os valores são iguais e 0 quando são diferentes, estes valores ao serem comparados com o sinal *clock* vão resultar em vários “10” que correspondem ao “1” no código *Manchester*.

O sinal codificado é comparado com o sinal ENABLE invertido numa OR. Quando o ENABLE invertido se encontra a “0”, o valor de saída da OR corresponde aos dados enviados pelo sensor. Quando o ENABLE invertido passa a “1”, a saída da OR terá sempre o valor “1” (o sensor fica calado).

No entanto é de salientar que antes de serem enviados os dados é necessário enviar o ACK que também necessita de ser codificado. Neste caso a codificação é feita com recurso a uma OR na qual entra o sinal ACK negado e o sinal *clock*. Assim quando o ACK é “0”, vai entrar como “1” na OR e a saída desta porta lógica vai ser sempre “1”. Quando o ACK é “1”, vai entrar como “0” na OR e a saída desta será igual ao *clock*, ou seja, “10”, que corresponde ao valor “1” no código Manchester. É de salientar que o sinal ACK aparece sempre primeiro que os dados uma vez que o sinal ENABLE só toma o valor 1 depois do ACK voltar a 0.

Os sinais codificados entram na AND. Quando o ACK está a zero e o ENABLE é 1 a AND é sensível aos dados enviados pelo sensor e os dados são enviados - sinal Dout.

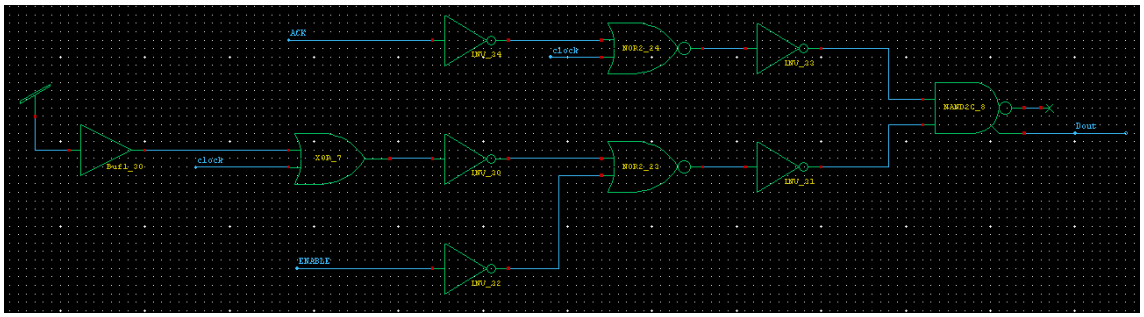

Figura 16 Codificador Manchester – sinal Dout

O sinal de saída Dout assemelha-se ao *busdata* a partir do momento em que o sistema está ativo.

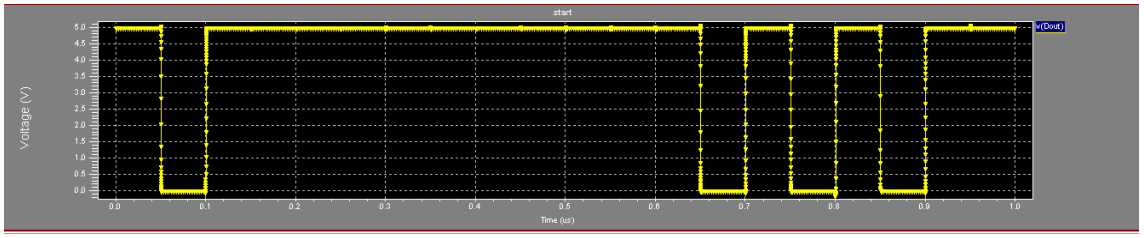

Figura 17 Sinal Dout

Deteção de Erros - Sinal ERROR

O sistema de deteção de erros permite detetar se há falta de sincronismo entre o *clock* e o *busdata* e faz a comparação entre estes dois sinais.

O sinal ACK e ENABLE entram numa NOR e a saída desta porta logica é “0” uma vez que eles são sempre diferentes. O sinal Dout é comparado com o *busdata* e sempre que forem iguais, ou seja, sempre que não haja erro, a saída da XNOR é “1”. Estas duas saídas vão ser aplicadas numa NOR que terá uma saída igual a “0” sempre que não houver erro ou “1” no caso do Dout não corresponder ao *busdata*.

Este sinal vai ser aplicado a dois *flip-flops* em paralelo, um com resposta na subida e outro na descida, o que permite sensibilidade aos 2 bits uma vez que há resposta a cada meio ciclo de relógio.

As saídas dos *flip-flops* são aplicadas numa OR o que resulta num sinal igual a “0” quando não há erro e num sinal igual a “1” quando há erro.

Este sinal ERROR é aplicado numa OR com o sinal Dout e assim se for o sinal ERROR for “0” os dados enviados pelo sensor são postos na linha, caso contrário, a saída vai ser “1” e a

linha vai ser colocada a “1”. Deste modo os dados enviados pelo sensor não vão aparecer na linha.

Há também a aplicação deste sinal na geração do sinal SET. Assim, no caso de existir um erro, o sinal ERROR vai provocar a mudança do sinal SET para “11” que por sua vez faz set aos *flip-flops*, colocando o *busdata* a “1” e terminando assim a comunicação.

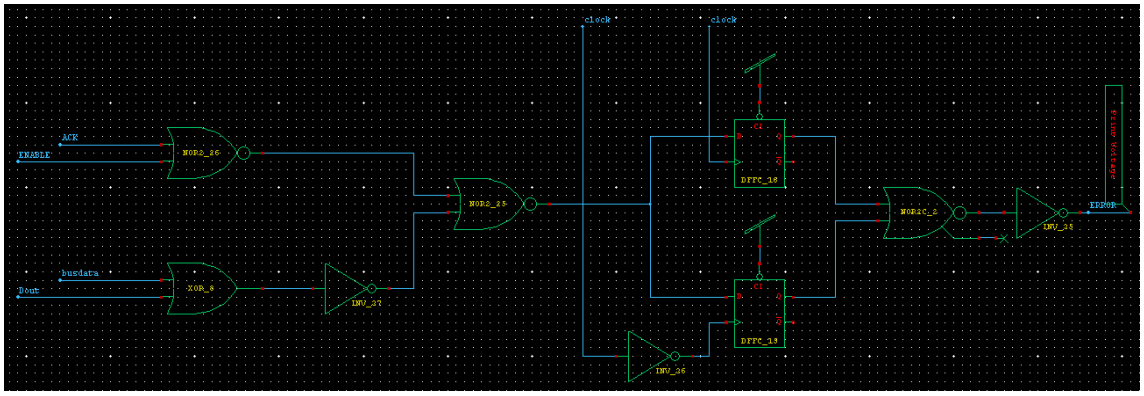

Figura 18 Circuito que gera o sinal ERROR

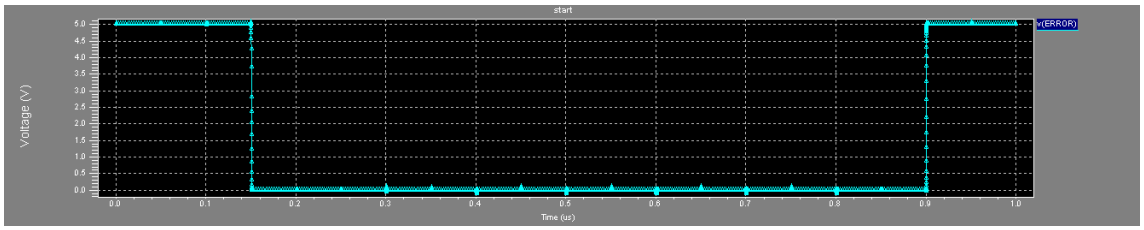

Figura 19 Sinal ERROR

Sinais Obtidos