

Experimento 8

FLIP-FLOPS:RS E JK

Lucas Mafra Chagas, 12/0126443

Marcelo Giordano Martins Costa de Oliveira, 12/0037301

¹Dep. Ciência da Computação – Universidade de Brasília (UnB)
CiC 116351 - Circuitos Digitais - Turma C

{giordano.marcelo, chagas.lucas.mafra}@gmail.com

Abstract. *In this experiment, we will study some types of flip-flops RS and JK by implementing, verifying their truth tables and comparing with the theoretical well-known results.*

Resumo. *Neste experimento, serão estudados os alguns tipos de flip-flops RS e JK, através de sua implementação e posterior verificação de suas respectivas tabelas verdade, comparando-as com os resultados esperados teoricamente.*

1. Objetivos

Apresentação do flip-flop como unidade armazenadora de memória. Observação do funcionamento e construção dos flip-flops RS, RS Gatilhado, SENHOR-ESCRAVO e JK SENHOR-ESCRAVO.

2. Materiais

- Pannel Digital
- *protoboard*
- Fios
- Portas Lógicas NAND, NOR e NOT.

3. Introdução

O flip-flop, ou multivibrador biestável, é um circuito digital pulsado capaz de servir como uma memória de um bit. Um flip-flop tipicamente tem dois sinais de entrada, o *SET* e o *RESET*, um sinal de clock (gatilho), um sinal de saída e o seu inverso. A pulsação ou mudança no sinal do clock faz com que o flip-flop mude ou retenha seu sinal de saída, baseado nos valores dos sinais de entrada e na equação característica do flip-flop. Existem vários tipos de flip-flops.

3.1. Flip-flop Latch RS

O valor guardado no flip-flop será mantido se *SET* e *RESET* forem ambos iguais a 0; irá mudar para 0, se a entrada *RESET* for 1, e se tornará 1 se a entrada *SET* for 1. O comportamento não será especificado se as duas entradas forem iguais a 1. Esse comportamento, neste relatório será referido como o estado proibido. Aqui temos uma implementação desse flip-flop feito apenas com portas NAND.

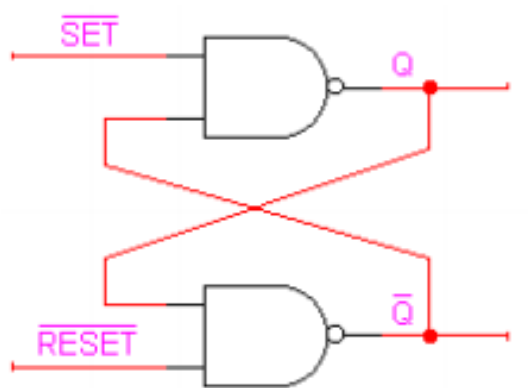


Figure 1. Latch RS

Para este tipo de implementação temos que considerar que a ativação de *SET* ou *RESET* é feita quando a entrada é 0. Além disso, temos que para esse flip-flop a saída possui um complemento, que sempre será o inverso da saída original.

3.2. Flip-Flop RS Gatilhado

Aqui há a inclusão de um gatilho (toggle) *T*. Quando houver variação do clock, o valor guardado no flip-flop será alternado ou mantido dependendo se o valor na entrada *T* (gatilho), se ele está em 1 ou 0. Se o valor de *T* é 1 temos que o valor será alterado de acordo com as entradas em *SET* e *RESET*. Se o valor de *T* é 0, o último valor alterado será armazenado, pois \overline{SET} e \overline{RESET} serão ambos iguais a 0. Portanto, a cada pulso de *T* temos o armazenamento da última mudança feita no flip-flop.

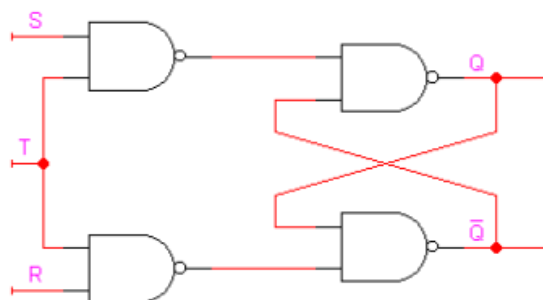


Figure 2. Latch RS Gatilhado

Para este flip-flop temos também a entrada proibida. Se ambos o *SET* e o *RESET* estiverem ativados, a saída será indeterminada.

3.3. Flip-Flop RS Gatilhado com PRESET e CLEAR

Este flip-flop é um Flip-Flop RS Gatilhado mas que pode forçar um resultado desejado na saída *Q*, já que as entradas *RESET* e *CLEAR* funcionam independentemente do gatilho *T* e das entradas *SET* e *RESET*.

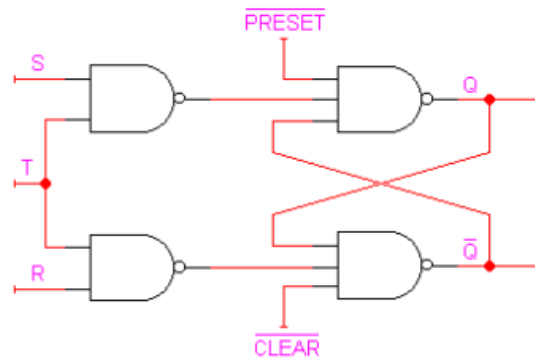


Figure 3. Latch RS Gatilhado com PRESET e CLEAR

3.4. Flip-Flop SENHOR-ESCRAVO

Este flip-flop é a junção de dois flip-flops RS Gatilhados. As saídas do primeiro flip-flop são as entradas do segundo. O gatilho T porém, nunca possui o mesmo valor nos diferentes flip-flops. Se o gatilho é 1 no flip-flop SENHOR, temos que os valores neste flip-flop podem ser alterados mas o flip-flop ESCRAVO, que terá seu gatilho em 0 permanecerá em repouso. Quando desligamos o gatilho do flip-flop SENHOR, acionamos o escravo, que pegará a informação armazenada no senhor e a armazenará também. Portanto, a cada pulso do gatilho temos que o valor no flip-flop SENHOR será alterado e essa alteração será transferida para o ESCRAVO.

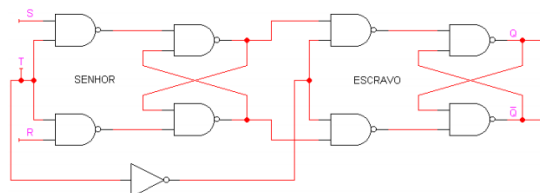


Figure 4. Flip-Flop SENHOR-ESCRAVO (Master-Slave)

3.5. Flip-Flop JK SENHOR-ESCRAVO

Este flip-flop permite a utilização do antes estado proibido. Temos que quando houver variação do clock, o valor guardado no flip-flop será alternado se as entradas J e K forem ambas iguais a 1 e será mantido se ambas forem iguais a zero. Se forem diferentes, haverá o *SET* ou o *RESET* dos valores.

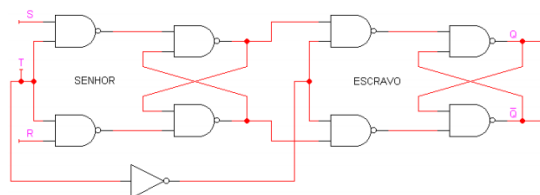


Figure 5. Flip-Flop JK SENHOR-ESCRAVO (Master-Slave)

4. Procedimentos

O relatório é dividido em 5 etapas:

- Montar o circuito Latch RS com portas NAND
- Montar o circuito Latch engatilhado
- Montar o circuito Latch engatilhado com entradas PRESET e CLEAR
- Montar o circuito de um flip-flop RS MESTRE-ESCRAVO usando portas NAND's
- Montar o circuito de um flip-flop JK MESTRE-ESCRAVO

Esse experimento é montado de forma gradual, assim um circuito está diretamente ligado ao outro.

4.1. Implementação de um circuito Latch RS com portas NAND

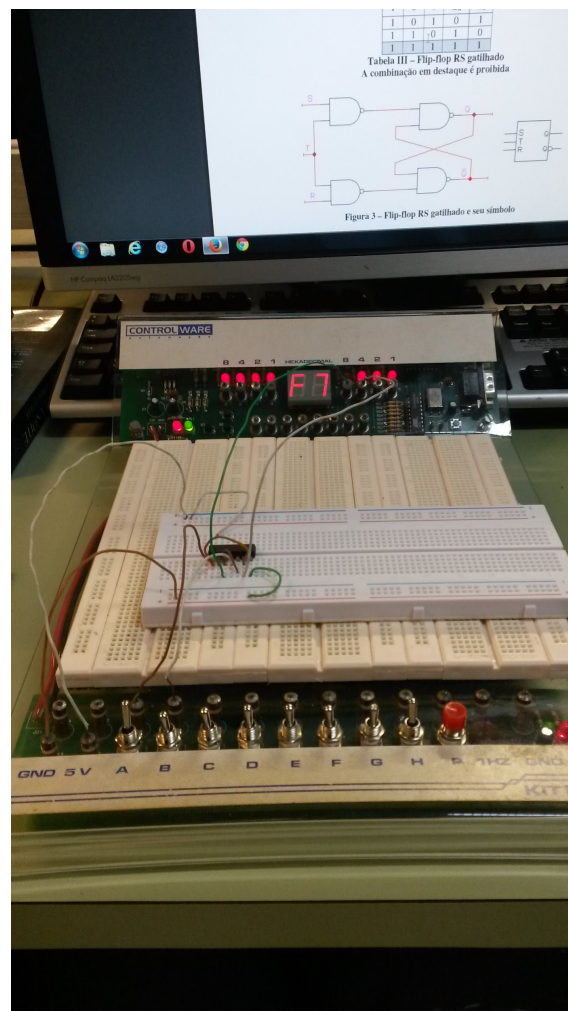


Figure 6. Latch RS utilizando portas NAND

Resultados obtidos para Q e \overline{Q} para a sequência 10,11,01,11,00,11 de \overline{SET} e \overline{RESET}

Observação: para este flip-flop temos que o \overline{SET} e o \overline{RESET} estão ativados em nível 0 ao invés de nível 1. Para as outras partes do experimento temos o caso oposto.

S	R	Q	\overline{Q}
1	0	0	1
1	1	0	1
0	1	1	0
1	1	1	0
0	0	1	1
1	1	0	1

Table 1. Tabela-Verdade do Latch RS

Vemos que quando \overline{RESET} está ativado, temos que a saída Q é igual a 0 e a saída \overline{Q} é igual a 1. Ao desligarmos o \overline{RESET} , deixando ambos o \overline{SET} e o \overline{RESET} em repouso, temos que o flip-flop mantém essa configuração das saídas armazenadas. Esse armazenamento também ocorre quando ativamos o \overline{SET} e desativamos o \overline{RESET} . A única diferença é que neste caso, Q é 1 e \overline{Q} é 0. Finalmente, quando ativamos ambos \overline{SET} e \overline{RESET} temos o estado proibido onde tanto Q como \overline{Q} são iguais a 1, o que é uma incoerência. Ao desligarmos as chaves após o estado proibido, temos Q e \overline{Q} apresentando um resultado qualquer. A partir disso, vemos que a tabela verdade obtida para este circuito é:

S	R	Q_{n+1}	\overline{Q}_{n+1}
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	\overline{Q}

Table 2. Tabela-Verdade do Latch RS

4.2. Implementação de um circuito Latch gatilhado

Vídeo 2.2

Vemos pelo vídeo que quando o gatilho T está desligado, ele armazena o último valor de Q e \overline{Q} que foram inseridos, independente do ativamento ou desativamento de \overline{SET} e \overline{RESET} . Antes de T estar desligado, tínhamos a chave RESET acionada, por isso que Q era 0 e \overline{Q} era 1. Já quando temos T ativado, ou seja, acionado em 1, temos que alterando SET e RESET obtemos os mesmos valores da tabela verdade em 4.1. Temos portanto que T define se o flip-flop está em repouso ou se ele está recebendo informação. Quando T é igual a 0 e o flip-flop está em repouso, temos que a saída referente à última entrada aplicada fica retida dentro do flip flop. Ficamos, portanto, com a seguinte tabela verdade:

T	S	R	Q_{n+1}	$\overline{Q_{n+1}}$
0	X	X	Q_n	$\overline{Q_n}$
1	0	0	Q_n	$\overline{Q_n}$
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

Table 3. Tabela-Verdade do Latch gatilhado


4.3. Implementação flip-flop RS SENHOR-ES CRAVO com portas NAND

Vídeo 2.3

Após construído o circuito, para cada possível entrada das chaves *SET* e *RESET* foi observada a saída do ESCRAVO após cada pulso aplicado em T. A análise foi feita a partir de pulsos pois para este tipo de circuito quando temos T=0, temos a informação sendo passada do flip-flop SENHOR para o flip-flop ESCRAVO e quando T=1 temos que o flip-flop ESCRAVO aguarda enquanto alteramos valores no flip-flop SENHOR, e queríamos fazer a observação do que o flip-flop armazenava após uma mudança (um pulso) completo. A partir disso, constrói-se a seguinte tabela verdade:

T	S	R	Q_{n+1}	$\overline{Q_{n+1}}$
0	X	X	Q_n	$\overline{Q_n}$
pulse	0	0	Q_n	$\overline{Q_n}$
pulse	0	1	0	1
pulse	1	0	1	0
pulse	1	1	1	1

Table 4. Tabela-Verdade do Flip-flop SENHOR-ES CRAVO

pulse = 

Vemos pela tabela e pelas imagens que o escravo armazenava o valor após o fim do pulso, alterado quando T = 1. Temos portanto, que a construção deste flip-flop teve sucesso.

4.4. Implementação de um flip-flop JK SENHOR-ES CRAVO com portas NAND

Com base na imagem abaixo e reutilizando os flip-flops construídos anteriormente, devia-se montar o circuito de um flip-flop JK SENHOR-ES CRAVO.

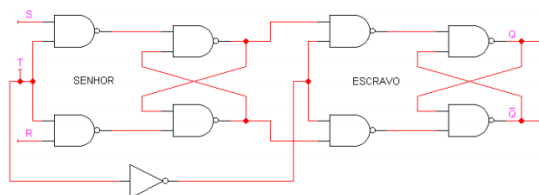


Figure 7. Flip-Flop JK SENHOR-ES CRAVO (Master-Slave)

E sua tabela deveria ser:

Entradas			Saídas	
T	S	R	Q_{n+1}	\overline{Q}_{n+1}
0	X	X	Q_n	\overline{Q}_n
\perp	0	0	Q_n	\overline{Q}_n
\perp	0	1	0	1
\perp	1	0	1	0
\perp	1	1	\overline{Q}_n	Q_n

Figure 8. Tabela verdade do Flip-Flop JK SENHOR-ESCRAVO (Master-Slave)

Mas, infelizmente, por conta de alguns fios com problemas, não conseguimos finalizar essa parte.

5. Análise dos Resultados

Analisando os dados da sessão 4.1 à 4.3, conseguimos comparar as tabelas verdades resultantes do experimento com as tabelas fornecidas pelo roteiro do experimento 8, mesmo que não tenhamos conseguido terminar a parte 4.4, a maior parte dos resultados está dentro dos conformes, portanto, consideramos o experimento um sucesso.

6. Conclusão

Neste experimento foi possível perceber a importância de um flip-flop na utilização de armazenamento de informações. Vimos que, se construídos corretamente, os flip-flops podem ser úteis em diversas aplicações

Auto-Avaliação

1. d
2. a
3. d
4. d
5. c
6. a
7. a