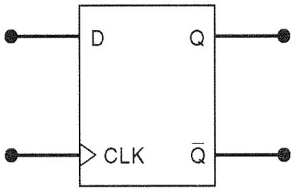
Elementos de Armazenamento

Vimos anteriormente que os sistemas microprocessados precisam de memória para armazenar os programas e os dados relativos a esses programas. Dentro do µP não é diferente, ele necessita de memória interna para poder armazenar os dados que vêm do exterior para poderem ser modificados (processados).

O Flip-Flop tipo D

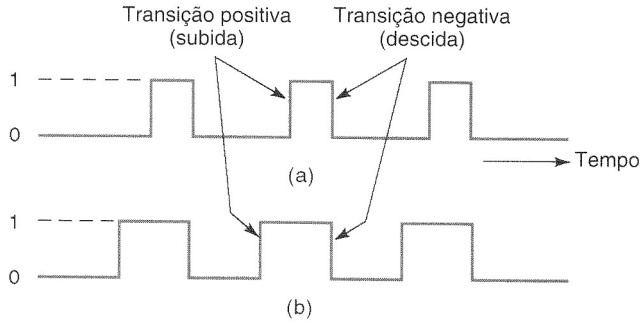
Dentro do µP, a memória tem que ser a mais rápida possível para que não cause demora na execução das instruções e por isso o Flip-Flop D (FFD) é escolhido como circuito de armazenamento básico (1 bit). Este tipo de memória se chama estática, diferente da memória dinâmica encontrada nas memórias RAM (fora do µP). Um FFD pode armazenar um bit indefinidamente enquanto a alimentação do circuito não for cortada.

O FFD tem uma entrada de dados D, uma entrada de controle CLK (clock) e duas saídas: Q (valor armazenado no FF) e $\overline{Q}$, sendo a segunda o inverso da primeira saída (o que é 0 vira 1 e o que é um vira 0). A figura abaixo mostra a representação de um FFD



Flip-Flop D com suas entradas e saídas

Um sinal digital de 1 bit pode assumir os valores 0 e 1 e é razoável portanto concluir que ao se passar de um valor a outro, o intervalo da transição deve ser o mais breve possível. Por isso os sinais representados são sempre com transições verticais, mesmo sabendo que isso não é possível na natureza. A figura abaixo detalha duas formas de onda digitais e mostra o que é uma transição positiva (ou de subida) e uma transição negativa (ou de descida).



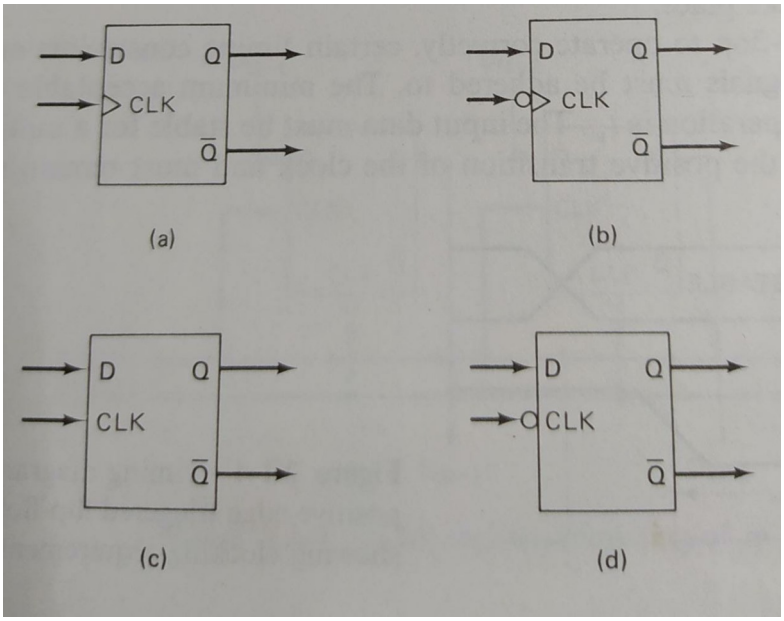
Forma de onda digital e suas transições

O FFD pode ser classificado quanto ao disparo do CLK:

1 – Nível
    1.1 – Alto
    1.2 – Baixo

2 – Transição
    2.1 – Subida
    2.2 – Descida

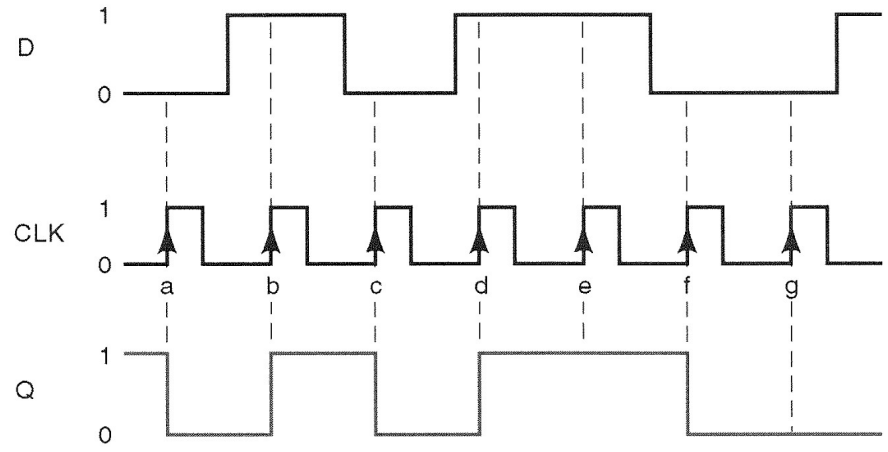
Quando os FFD são disparados por nível, eles são também chamados de ***latch***. A figura abaixo mostra a representação dos 4 tipos de FF segundo a classificação feita acima:



Símbolos dos tipos de FFD quanto ao disparo:
a) disparo por transição de subida; b) disparo por transição de descida;
c) disparo por nível alto; d) disparo por nível baixo

Funcionamento:

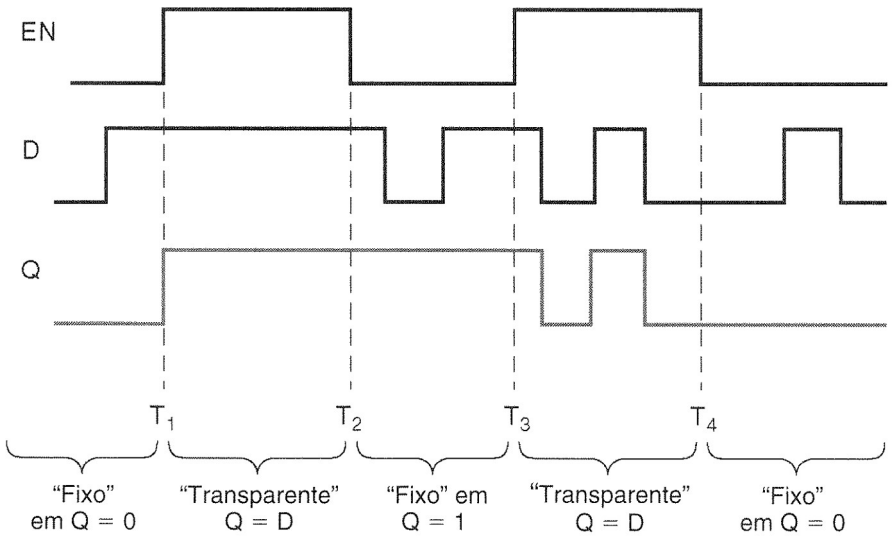
Um FFD disparado por transição armazena (fica disponível em Q) o valor da entrada D durante a transição de disparo. Como exemplo, usando um FFD disparado por transição de subida, A entrada D só é armazenada durante a transição de clock de 0 para 1. Fora este momento em que a entrada D pode ser armazenada, o FFD mantém o último valor sem alteração. A figura abaixo mostra esta situação. Perceba que são marcadas 7 transições de subida (***a*** a ***g***) e que só é possível mudanças na saída Q nos momentos das transições positivas.



Exemplo de funcionamento do FFD disparado por transição de subida

O FFD disparado por transição de descida funciona de forma parecida, porém a entrada D será armazenada somente durante a transição de descida do *clock*. O *clock* portanto controla a entrada de dados.

O FFD disparado por nível (ou *latch*) armazena o valor de entrada enquanto o nível de disparo durar. Exemplificando com um FFD disparado por nível alto, a figura abaixo mostra que a saída Q segue a entrada D (momento transparente) sempre o nível do *clock* (no caso do *latch* também pode ser chamado de *enable* – habilita e representado por EN) estiver em 1 e armazena o último valor (fixo) sempre que o EN ou *clock* for 0.



Exemplo de *latch* disparado por nível alto

De maneira análoga o FFD disparado por nível baixo deixa a entrada transparente quando o *clock* estiver em 0 e fixo quando o *clock* estiver em 1.

É importante fazer esta distinção entre os disparos pois veremos que há eventos que acontecem de acordo com nível e com transição em um sistema microprocessado.

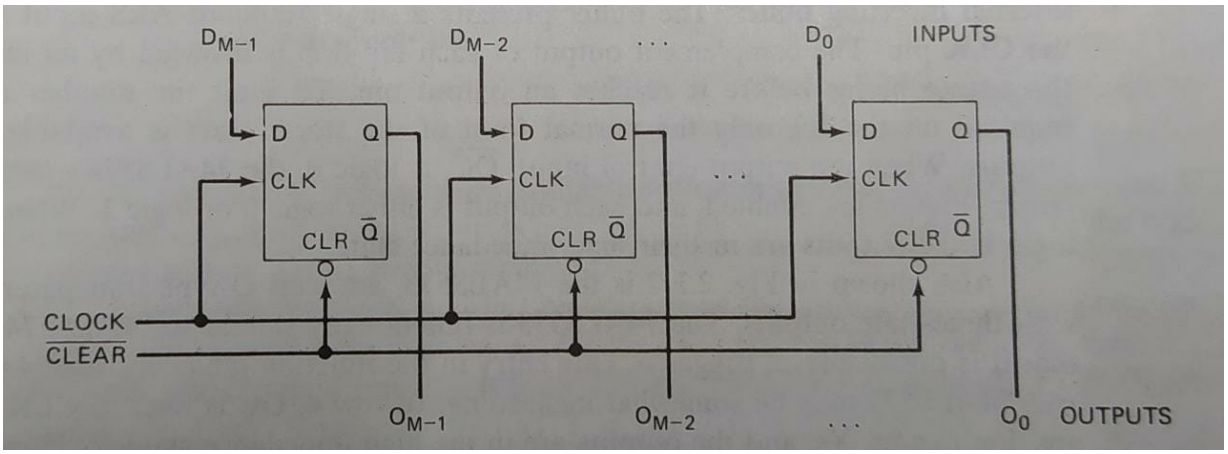
Existem entradas assíncronas de CLEAR e PRESET que zeram e colocam em 1 o FFD independente do controle CLK. Desta forma, podemos garantir um valor específico em um determinado instante do circuito ao acessar essas entradas pois quando se liga um FFD, podemos ter um valor binário indeterminado.

Registradores de m bits

Para armazenar vários bits de dados simultaneamente, podemos ligar o mesmo *clock* em vários FFD, desta forma, estaremos conectando-os de forma paralela. Quando um bit for armazenado em um deles, todos os outros FFDs estarão armazenando seus bits, pois o controle é o mesmo. A figura abaixo mostra um registrador de *m* bits formado por *m* FFDs.

É bom notar que os valores armazenados pelos FFDs podem diferir uns dos outros porém o momento em que eles são armazenados deve ser o mesmo para todos eles.

A operação de armazenar um valor é chamada de **escrita** e esta operação destrói o valor anterior. A operação de se obter um valor armazenado é chamado **leitura** e é uma cópia daquele valor, não sendo uma operação destrutiva.



Registrador de *m* bits

Os m bits de dados armazenados num registrador formam uma **palavra**. Uma palavra é simplesmente um número contíguo de bits que pode ser operado pelo *hardware* como um grupo. O número de bits m é chamado de **largura da palavra**. 4 bits de informação tem o nome de ***nibble*** e 8 bits tem o nome de ***byte***.