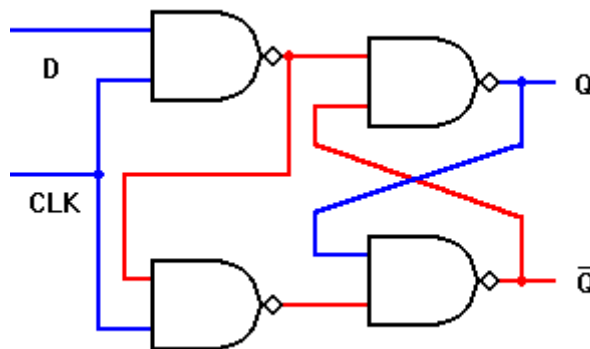


# Trabalho de Lógica Matemática

## Objetivo

Implemente um programa em C ou C++ que **simule a lógica de um clocked d latch ou gated d flip flop**. Esse dispositivo eletrônico é usado para implementar a lógica de armazenamento de um bit em um circuito e usa apenas as operações lógicas de conjunção (AND) e negação (NOT) através da porta lógica NAND (AND seguido de uma negação - NOT). O programa deverá receber dois valores lógicos D e CLK como entrada e retornar o valor Q correspondente, conforme a ilustração abaixo.



## Como funcionará o programa

Os seguintes passos devem ser seguidos na sua implementação:

1. Em primeiro lugar pesquise sobre o **latch** em livros de lógica, circuitos e na internet. Tente entender exatamente o problema que está sendo resolvido antes de partir para a prática.
2. Crie uma função para representar um **latch para apenas um bit**. A função deverá receber como entrada os parâmetros D, CLK e o valor Q anterior. Ela deverá retornar o valor do bit Q correspondente àquela chamada. Realize o calculo do valor Q **usando apenas operações binárias** das linguagens C e C++.
3. Solicite ao usuário para digitar um valor numérico decimal inteiro.
4. Mostre o número digitado pelo usuário em base binária com até 8 bits.
5. Chame 8 vezes a função (uma para cada bit) para armazenar o valor digitado pelo usuário. Use CLK=1, Q=0 e o valor D correspondente a cada um dos bits do número.
6. Faça novamente 8 chamadas à função (uma para cada bit) com CLK=0 e o Q correspondente para obter o valor armazenado e mostrá-lo ao usuário.

## Exemplo de execução

Valor digitado pelo usuário: 25

Valor em binário: 00011001

Chamadas à função:

- latch(CLK=1, Q=0, D=0)
- latch(CLK=1, Q=0, D=0)
- latch(CLK=1, Q=0, D=0)
- latch(CLK=1, Q=0, D=0)
- latch(CLK=1, Q=0, D=0)
- latch(CLK=1, Q=0, D=0)
- latch(CLK=1, Q=0, D=0)

Retorno das chamadas: 00000000

Chamadas à função:

- latch(CLK=1, Q=0, D=0)
- latch(CLK=1, Q=0, D=0)
- latch(CLK=1, Q=0, D=1)
- latch(CLK=1, Q=0, D=1)
- latch(CLK=1, Q=0, D=0)
- latch(CLK=1, Q=0, D=0)
- latch(CLK=1, Q=0, D=1)

Retorno das chamadas: 00011001

Chamadas à função:

- latch(CLK=0, Q=0, D=0)
- latch(CLK=0, Q=0, D=0)
- latch(CLK=0, Q=1, D=0)
- latch(CLK=0, Q=1, D=0)
- latch(CLK=0, Q=0, D=0)
- latch(CLK=0, Q=0, D=0)
- latch(CLK=0, Q=1, D=0)

Retorno das chamadas: 00011001

## O que é necessário saber

Utilize operações lógicas para realizar as operações (e, ou, ou exclusivo, não). **Não poderá ser utilizada nenhuma operação aritmética (+, -, \*, /) nem estruturas condicionais (if, else) e de laço (for, while, do) para encontrar o resultado do valor Q.**

Para realizar essa tarefa você deverá conhecer bem as operações lógicas dentro da linguagem de programação. Geralmente trabalhamos com operadores lógicos de conjunção e disjunção nestas linguagens:

- **&& (e)**: realiza a operação de conjunção entre dois valores inteiros.
- **|| (ou)**: realiza a operação de disjunção entre dois valores inteiros.
- **! (negação)**: inverte o valor lógico de uma variável.

Mas, neste programa vamos precisar dos **operadores lógicos binários**:

- **& (e binário)**: realiza a operação binária de conjunção entre dois valores. É diferente da operação de conjunção porque aqui ela é realizada em cada um dos bits, produzindo um número binário que corresponde ao resultado da conjunção bit a bit.

Ex: (0b10001 & 0b11011) produz o binário 0b10001.

- **| (ou binário)**: realiza a operação binária de disjunção entre dois valores. É diferente da operação de disjunção porque aqui ela é realizada em cada um dos bits, produzindo um número binário que corresponde ao resultado da disjunção bit a bit. Ex: (0b10001 | 0b11011) produz o binário 0b11011.
- **~ (negação binária)**: inverte o valor de todos os bits de um número. ~(0b010) produz o binário 0b1111101.
- **<< (bitshift left)**: desloca todos os bits à esquerda pelo número de vezes especificado. Ex: 0b00010 << 1 produz o binário 0b00100.
- **>> (bitshift right)**: desloca todos os bits à direita pelo número de vezes especificado. Ex: 0b00010 >> 1 produz o binário 0b00001.

## Sugestões

Algumas sugestões para realizar a implementação:

- Lembre-se que o **latch** utiliza um circuito lógico em *loop* para armazenar cada bit (Q). Como não temos meio de reproduzir esse comportamento, o Q que corresponde ao valor armazenado será passado como parâmetro da função. O programa vai apenas **simular** o comportamento lógico e o armazenamento do valor.
- Entenda bem como funcionam os operadores binários (e, ou, not, bitshift) antes de começar este trabalho. Eles serão necessários para concluir essa tarefa.
- Trabalhe com o tipo de dados unsigned char, que contém apenas 8 bits. Não deixe de usar modificador unsigned para evitar problemas.
- Implemente funções úteis antes de tentar resolver o problema principal: função de extrair um bit em determinada posição, função de mostrar um número inteiro como binário, função que soma dois operandos e um “vai um”, etc...

## Avaliação

As informações apresentadas nessa instrução podem não ser suficientes para implementar o programa. Em caso de dúvidas, cabe aos estudantes pesquisarem sobre o assunto e consultarem o professor.

Os estudantes deverão entregar o código fonte do programa (arquivos .c, .cpp, .h) que implementa o que foi pedido. As notas serão individuais e o professor poderá solicitar aos estudantes que apresentem o trabalho individualmente.

Trabalhos copiados terão a nota de ambos alunos zerada.

Os trabalhos serão avaliados nos seguintes critérios:

- Verificação da corretude da funcionalidade principal e funcionalidades acessórias.
- Estruturação do código em funções para facilidade de correção.
- Clareza nas informações apresentadas para o usuário e codificação (nomes de variáveis, funções, comentários, etc).
- Como atividade extra, você pode trabalhar com números maiores: 64 bits ou 32 bits.