

# Introdução aos Sistemas Lógicos - DCC114/ TZ - 2022/2

## Trabalho Prático 2: Multiplicador Binário Sequencial

**Professor:** Marcos Vieira ([mmvieira@dcc.ufmg.br](mailto:mmvieira@dcc.ufmg.br))

**Monitor:** Paulo Mol ([pmolufmg@gmail.com](mailto:pmolufmg@gmail.com))

Thaís Ferreira da Silva - 2021092571

### 1. Breve Descrição

Aprendemos em sala de aula que uma das operações mais custosas é a multiplicação, que pode ser construída através de consecutivas somas binárias. Então para esse trabalho, implementei um multiplicador binário sequencial (mbs) 8x8 usando como referência a figura 2 e 3 do enunciado. Escolhi o primeiro modelo por parecer mais simples e intuitivo de construir.

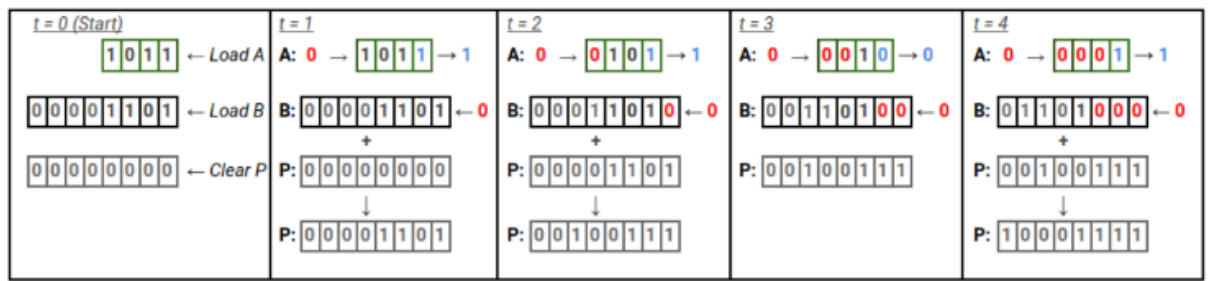


Figura 2. Multiplicador sequencial

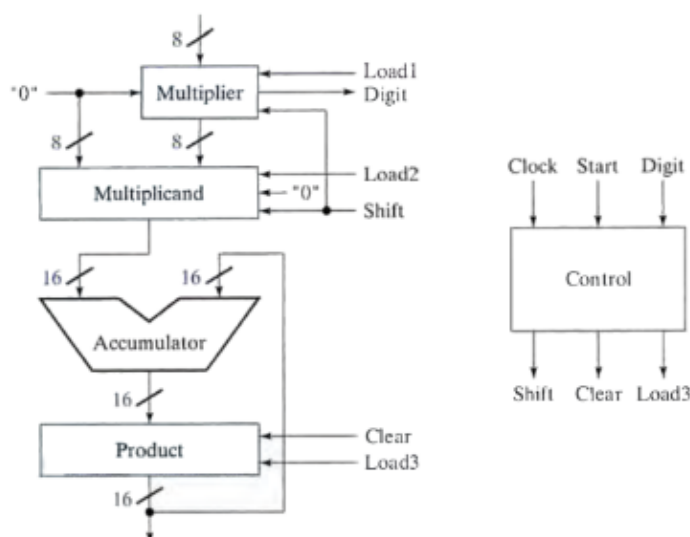


Figura 3. Multiplicador sequencial

Sobre a implementação do código. No início foi um pouco complicado converter a ideia para o verilog, mas o processo ficou um pouco mais claro depois que entendi melhor o multiplicador sequencial. O módulo mbs recebe como solicitado o Clock, Start, Multiplicando, Multiplicador e Produto, mas decidi modificar os nomes para que o código ficasse mais parecido com a figura 2. Então modifiquei as variáveis para Clock, Start, B, A e P.

Depois criei 3 novas variáveis auxiliares para todo o programa, um para o contar quantos deslocamentos foram realizados (integer count), e dois registros para o A e

B, tendo em vista que recebi eles apenas como variáveis input e que o B precisava mudar de 8 bits para 16 bits devido aos deslocamentos.

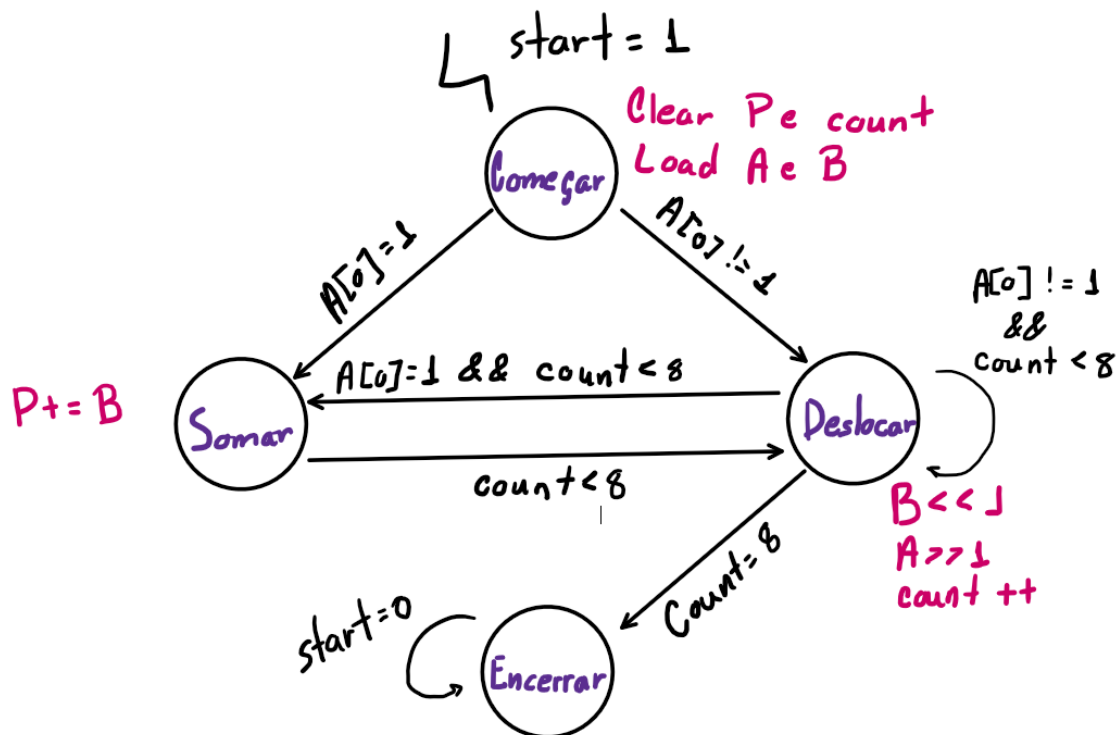
Depois disso, o código entra em um loop que obedece a borda de subida do Clock. Quando o Start é 1, o programa inicializa as variáveis auxiliares, e zera o valor do produto. Lembrando que é importante completar os 8 bits mais significativos de B (Multiplicando) com zeros para que haja espaço livre para o deslocamento.

Quando o Start é 0, ou seja, pras outras instâncias de tempo, verificamos se o bit menos significativo de A é 1 e nesse caso somamos B ao produto P. Depois sempre deslocamos o B 1 bit para a esquerda, e o A 1 bit para a esquerda.

Já no testebench.sh inicializei os registradores Clock, Start, B, A e P, chamei a função mbs explicada anteriormente, e implementei um clock alternando o seu valor entre 0 e 1 a cada segundo (isso ocorre 20 vezes que corresponde a 4 multiplicado pelo número de casos de teste). Por fim, realizei 5 casos de teste escolhidos aleatoriamente que estão explicitados na parte de Formas de Ondas.

## 2. Diagrama FSM

Nessa máquina de estados existem 4 estados possíveis, começar a multiplicação, somar o multiplicando ao produto, deslocar o multiplicando e multiplicador, e encerrar a multiplicação.

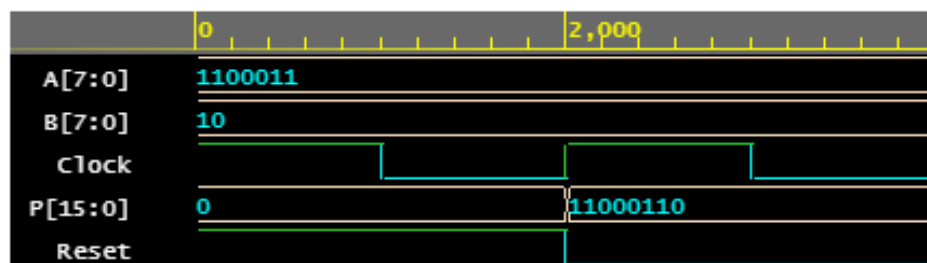


### 3. Forma de Onda

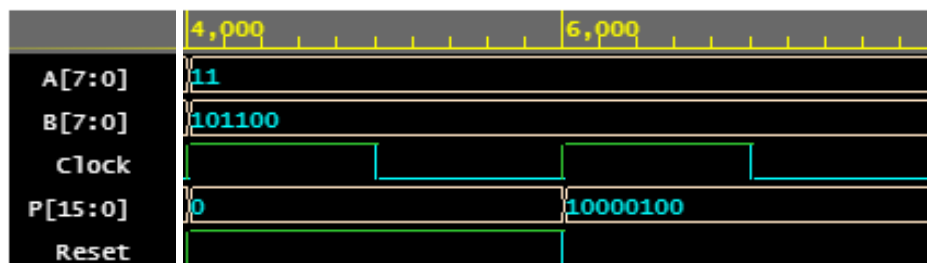
Os valores obtidos na forma de onda correspondem a 5 multiplicações:

Multiplicação (A*B)		Resultado	
99*2	198	1100011*10	11000110
3*44	132	11*101100	10000100
69*24	1656	1000101*11000	11001111000
80*100	8000	1010000*1100100	1111101000000
32*200	6400	100000*11001000	1100100000000

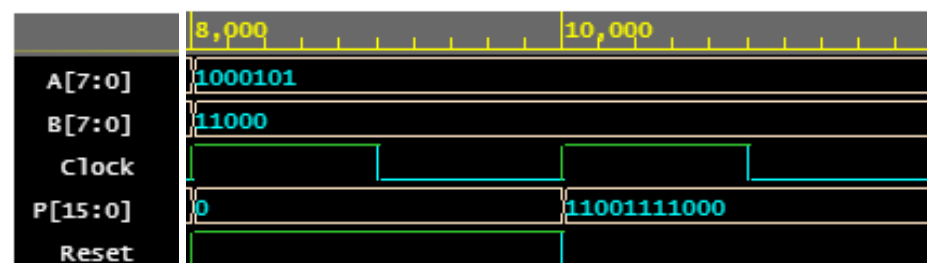
$$99*2 = 198$$



$$3*44 = 132$$



$$69*24 = 1656$$



$$80*100 = 8000$$

