

Programa para Excelência em Microeletrônica

Módulo: Nivelamento

Matéria: Sistemas Digitais

Desenvolvimento de um Microcomputador

Aluno: Gustavo Vilar de Farias

SONY
make.believe



FUNETEC-PB
Fundação de Educação Tecnológica
e Cultural da Paraíba



Sumário

[Introdução](#)

[Teoria](#)

[Desenvolvimento do Projeto](#)

[Contador do Programa](#)

[Acumulador](#)

[REM](#)

[Somador 8 bits](#)

[Subtrador 8 bits](#)

[Somador/Subtrador](#)

[Memória](#)

[Registrador B](#)

[Registrador de Instruções](#)

[Registrador de Saída](#)

[Contador em Anel](#)

[Ciclo de Busca](#)

[Ciclo de Execução](#)

[Controlador](#)

[Registrador de Saída](#)

[Conclusões](#)

Introdução

Uma calculadora é uma ferramenta essencial na vida de várias pessoas sendo uma das funcionalidades básicas de todo computador e venho por meio deste relatório mostrar como funciona e como foi feito um microprocessador capaz de realizar as funções de soma e subtração.

Teoria

O microprocessador uP é dividida em três áreas, nove sub-blocos e sua função principal é a de soma e subtração.

Cada sub-bloco tem uma função diferente, mas são todos interligados e interdependentes.

O microprocessador trabalhar com comandos guardados na *RAM* que são executadas de acordo com o contador do programa, onde vai incrementando e trocando o comando a ser executado.

Para sincronizar tudo existe o controlador, que toma conta do clock, e dos sinais dos outros blocos de acordo com a instrução que é lhe dada.

Desenvolvimento do Projeto

Para a implementação do projeto, resolvi fazer cada bloco por vez para ficar mais fácil o reuso e a solução de problemas, explico como foi feito cada bloco na seção abaixo.

Contador do Programa

Para o contador do programa, foi usado *flip-flops JK* onde a saída Q de um *flip-flop* era o clock do *flip-flop* seguinte, o J e o K são ligados no VCC sempre, para poder ocorrer a troca de sinais quando obtiver o pulso de clock e assim poder ir contando, lembrando que o pulso de clock só passa quando Cp estiver ativo e a saída só vai para o barramento quando Ep estiver em estado lógico alto.

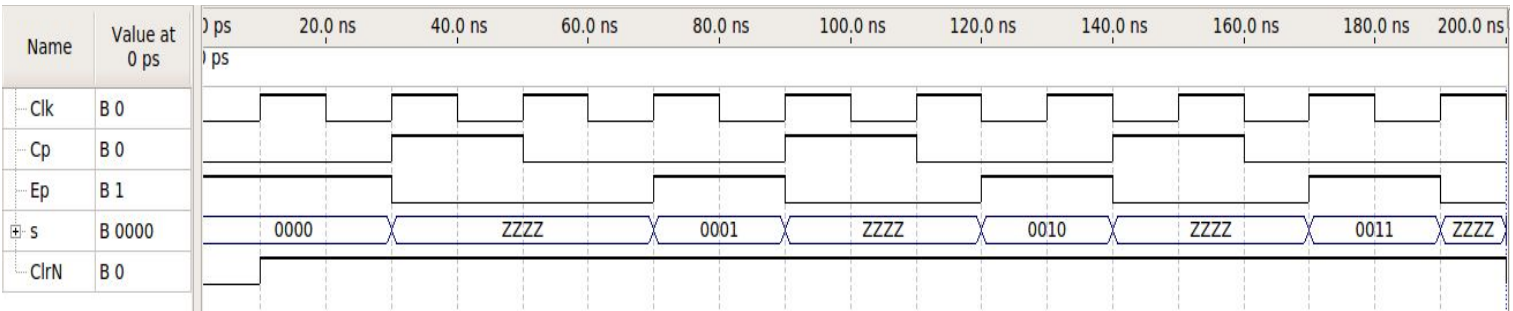


Figura 1 - Testes do Contador do Programa

Acumulador

No acumulador os *flip-flops D* serviram para o armazenamento dos dados, sempre quando ocorrer um pulso de clock e o sinal de La for baixo o que estiver no barramento é colocado nas saídas Q dos respectivos *flip-flops*, e a entrada Ea controla os *tribuffers*,

para permitir ou não a saída dos dados no barramento.

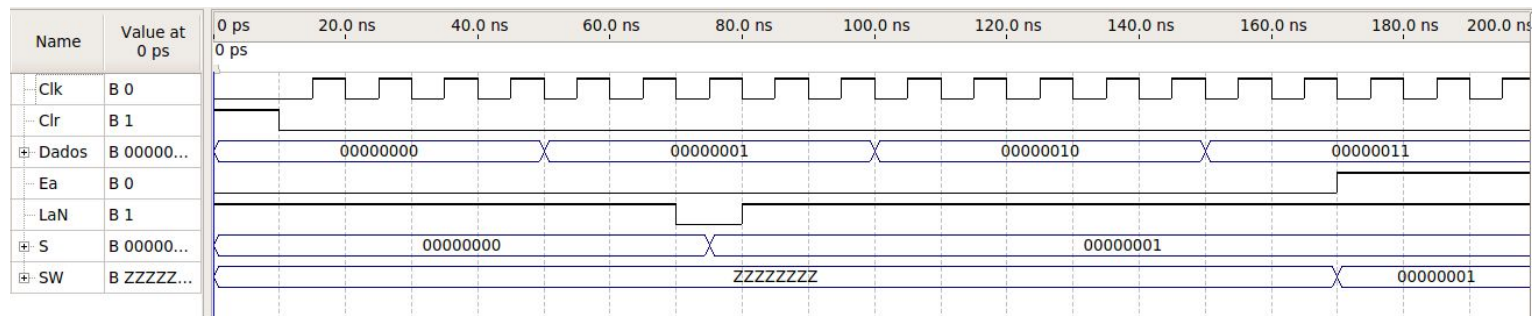


Figura 2 - Testes do Acumulador

REM

O esquema do REM é o mesmo do acumulador a diferença é a quantidade de bits da saída e entrada, e a saída vai para a memória e não para o barramento, também não existe uma entrada para controlar a saída dos dados, ou seja, quando o dado entra ele já vai direto para saída

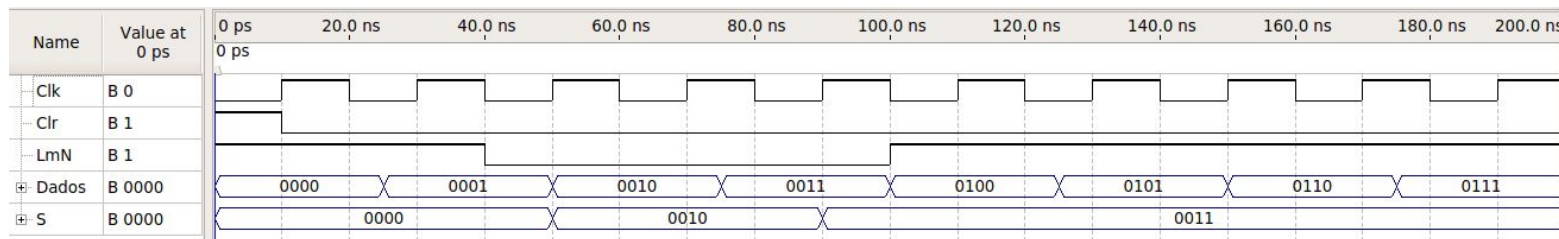


Figura 3 - Testes na REM

Somador 8 bits

Primeiramente foi feito o meio-somador que apenas somava um bit com o outro e retornava o resultado e o “resto”, para fazer a soma utiliza-se uma *ou exclusiva* e para o “resto” usa-se uma *and* com os 2 bits para serem somados.

Logo após o somador completo foi feito, nesse além de somar 2 bits ainda aceitava a entrada do “resto”, primeiro é somado os 2 bits, depois somamos a saída da primeira soma com o “resto” de entrada ai obtemos a soma e o “resto” de saída é uma *ou* com o “resto” das 2 somas.

Para finalizar juntamos somadores completos e temos um somador de 8 bits.

Subtrador 8 bits

Com o subtrador foi bem semelhante, a diferença é que no meio subtrador usamos uma *not* na primeira entrada da *and* para darmos o “resto” e no subtrador completo o resultado da subtração dos bits é o minuendo da subtração com o “resto”, as outras coisas são totalmente semelhantes ao Somador.

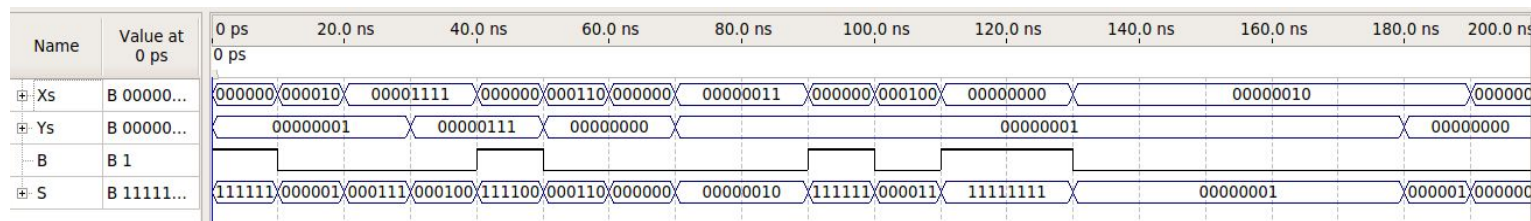


Figura 4 - Testes do Subtrador

Somador/Subtrador

Para fazer esta junção mando as 2 entradas para o Somador e para o Subtrador e uso uma espécie de MUX para selecionar qual saída vai para o barramento, o do Somador quando Su for baixo e o do Subtrador quando Su for alto, a saída também só irá para o barramento quando Eu for alto.

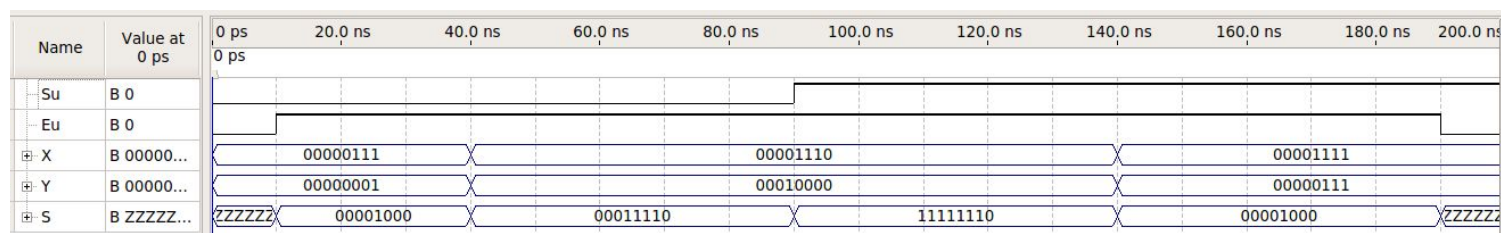


Figura 5 - Testes do Somador/Subtrador

Memória

No armazenamento dos dados utilizei uma memória ROM com um barramento de endereço de 4 bits e um barramento de saída com 8 bits, os dados foram guardados em um arquivo .hex que representa os comandos a serem executados no uC e as informações a serem processadas. Esses dados só serão apresentados ao barramento quando o estado de Ce for baixo.

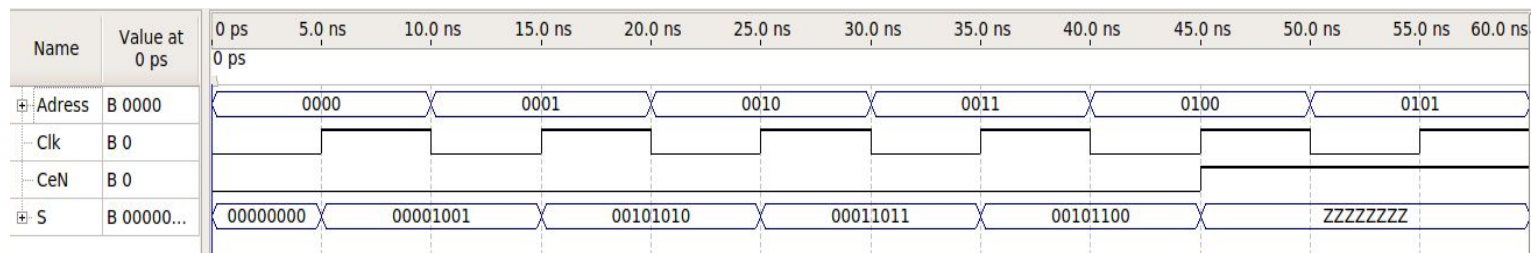


Figura 6 - Testes na Memória

Registrador B

Este sub-bloco foi feito idêntico ao Acumulador só tendo como diferença a saída que é apenas uma.

Registrador de Instruções

Já no registrador de instruções usamos o mesmo esquema dos *flip-flops D* para guardar o que existe no barramento quando Ln for baixo, mas para as saídas dividimos ela em 2 nibbles o que tem os 4 bits mais significativos vai para o controlador e o que tem os 4 bits menos significativos vai para o barramento quando o sinal de Ei for baixo.

Registrador de Saída

Este registrador tem a única função de guardar em *flip-flops D* o que se encontra no barramento quando *Lo* for baixo e jogar na saída do microprocessador.

Contador em Anel

Um sistema de deslocamento de bits foi usado na fabricação do contador em anel, quando o sinal de *Clr* for baixo seto o primeiro *flip-flop D* e reseto os demais, barra a entrada de clock para ser detectado a borda de descida e a cada pulso de clock o bit é deslocado para o próximo *flip-flop*, o último *flip-flop* é ligado ao primeiro. Com isso o bit setado com 1 vai passando adiante até chegar ao último *flip-flop* que passa novamente para o primeiro, os bits resetados com 0 também vão se deslocando.

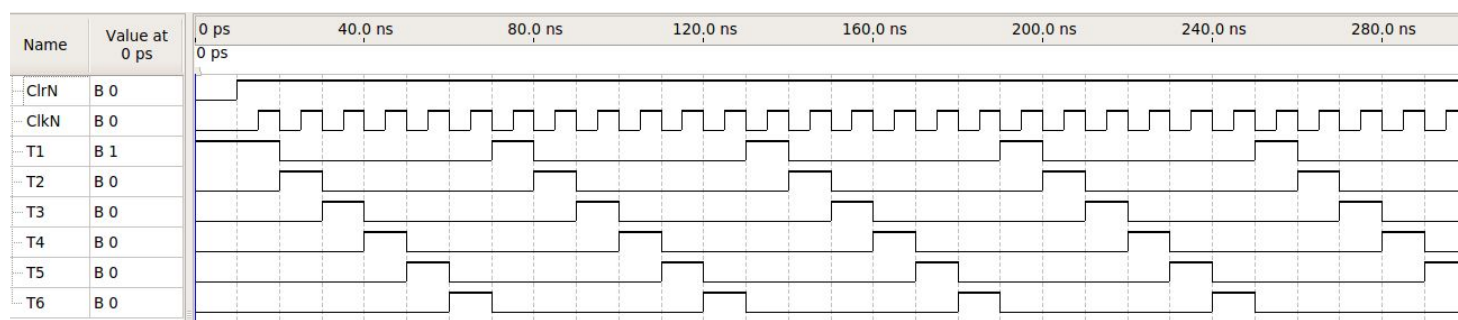


Figura 7 - Testes do Contador em Anel

Ciclo de Busca

No desenvolvimento do ciclo de busca procurei achar padrões nos sinais saída, consegui observar que as saídas *Ei*, *La*, *Lb* e *Lo* independente do ciclo de trabalho irão ficar com sinal alto, já as saídas *Ea*, *Su* e *Eu* vão sempre ficar com um sinal baixo, a saída *Cp* recebe um sinal alto apenas quando estivermos no ciclo de trabalho T2, as saídas *Ep* e *Lm* estarão ativas somente o ciclo de trabalho for o T1, e quando estivermos no ciclo de trabalho T3 ativamos a saída *Ce* e desativamos a *Li*.

Ciclo de Execução

Com o ciclo de execução busquei analisar e juntar as Rotinas e os ciclos de execução que fazem a mesma coisa, como o T4 é igual para todas as rotinas, o T5 e o T6 é igual nas rotinas de ADD e SUB diferenciando apenas o SU no ciclo T6 que irá ser ativo quando estiver na rotina de SUB, o T6 da rotina LDA é igual ao T5 e T6 da rotina OUT, para finalizar o *Lo* é ativado quando estivermos no ciclo T4 da rotina OUT. Após essa junção utilizei uma espécie de *MUX* para selecionar qual sinal passará quando estiver em determinado ciclo/rotina.

Controlador

No controlador fica reunido o Contador em Anel, Ciclo de Busca e Ciclo de Execução. O Contador em Anel nos dá T1, T2, T3, T4, T5 e T6 onde T1, T2 e T3 vão para o Ciclo de Busca e os demais vão para o Ciclo de Execução. Os sinais advindo do Registrador de Instruções passam por portas lógicas que ativam os sinais LDA, ADD, SUB, HLT ou OUT, estes sinais vão também para o Ciclo de Execução com exceção do HLT, que quando ativado barra a saída do clock. Para decidir qual sinal irá para as saídas uso uma espécie de “*Decididor*” que quando o sinal C for ativo vai para a saída

as entradas do Ciclo de Busca quando for baixo sai os sinais do Ciclo de Execução. C é ativo quando estivermos no ciclo T1, T2 ou T3.

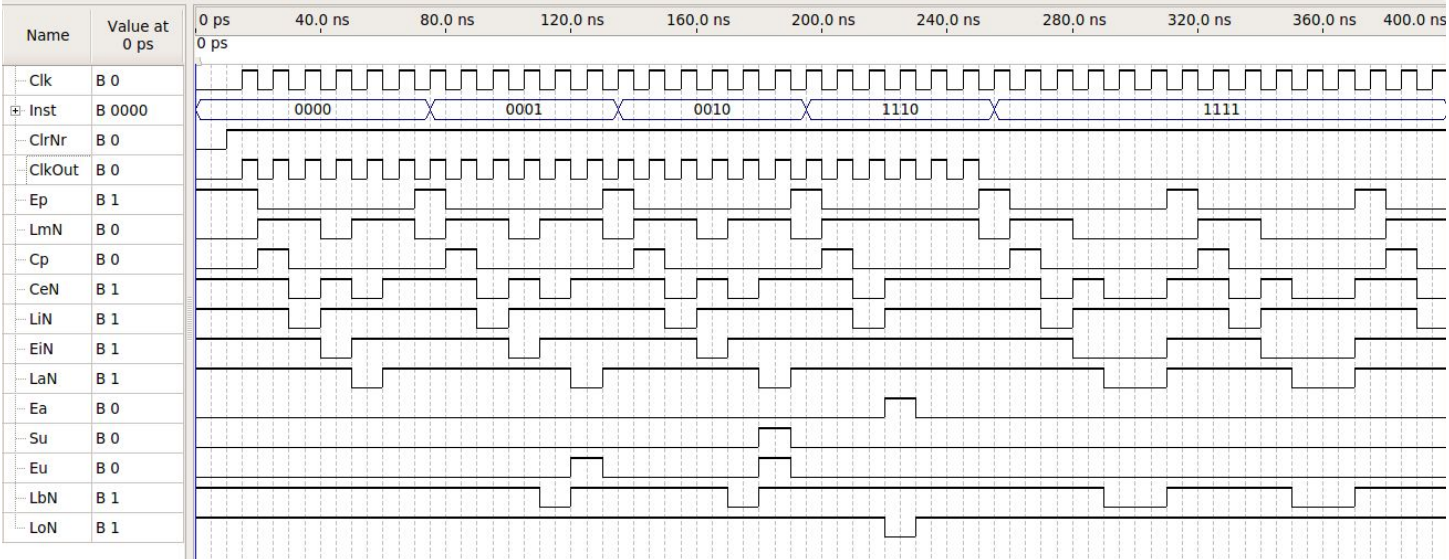


Figura 8 - Testes no Controlador

Registrador de Saída

Este registrador guarda o que estiver no barramento quando Lo for 0 e coloca imediatamente na saída.

Conclusões

Este trabalho me ajudou a entender ainda mais o funcionamento de um computador e com cada sub-bloco construído e testado o menor dos trabalhos foi fazer a junção dos mesmos, onde foi preciso apenas um barramento para comunicar os sub-blocos, ligar as saídas do Controlador com os sinais de entrada dos sub-blocos, colocar um sinal de Clk para limpar todos os blocos antes do início dos processos e algumas ligações necessárias entre blocos, como a saída do Acumulador e do Registrador B ligadas ao Somador/Subtrador, a saída do REM com a entrada de endereços da Memória, saída do Registrador de Instruções com o Controlador e a saída do Registrador de Saída com uma saída. As demais entradas e saídas de dados são ligadas ao barramento.