Ministério da Educação
Universidade Federal de Itajubá

ELTD15 – Laboratório de Sistemas Digitais – Diagramas ASM e de Estados Finitos

1– Objetivos

- Utilizar a linguagem de descrição de *hardware* Verilog;
- Utilizar o *software* ModelSim-Altera para simular os circuitos descritos em Verilog com TestBench;
- Praticar diagramas ASM, diagrama de estados finitos e suas conversões para circuitos.
- Desenvolver um pequeno sistema – Multiplicador – baseado no algoritmo soma e desloca.
- Entender como adicionar vários *modules* diferentes a uma hierarquia *Top no Quartus.*

2- O diagrama ASM abaixo mostra a diagramação de um multiplicador. O algoritmo em si é mostrado ao lado.

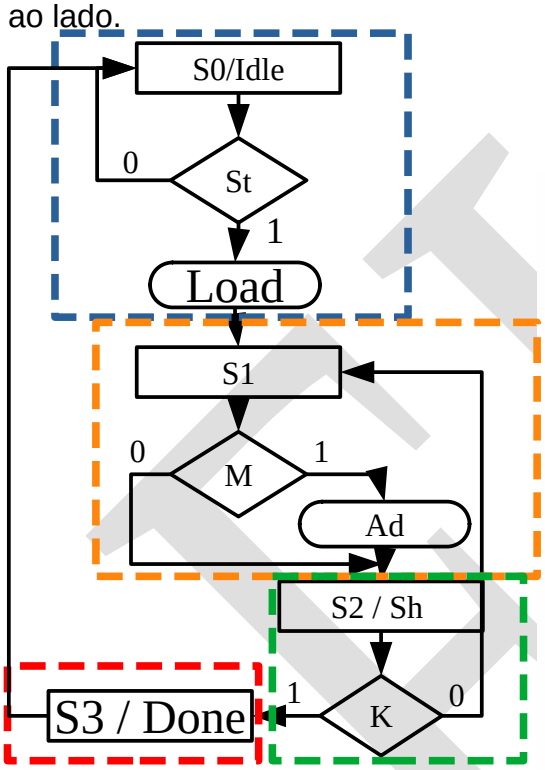

Figura 1: Diagrama ASM

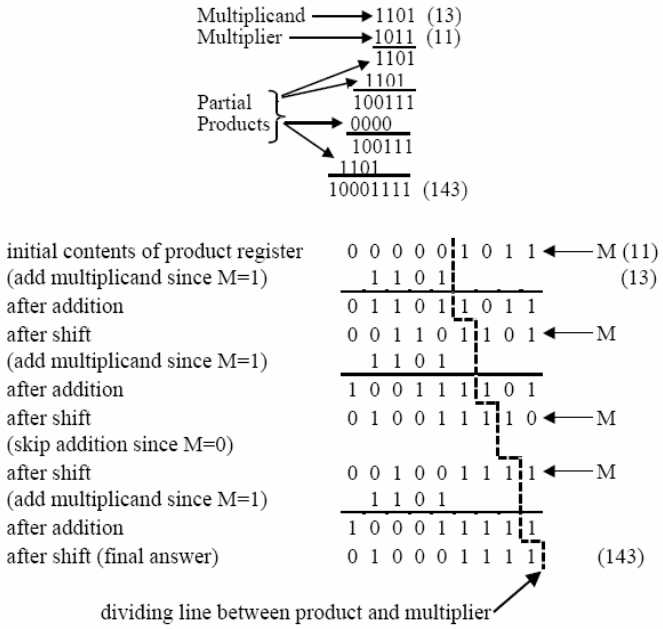

Figura 2: Algoritmo

Desenvolva a descrição verilog que implemente o diagrama ASM. Para isso, utilize descrições comportamentais para descrever os blocos (*cada módulo tendo seu próprio projeto para que cada um possa ser testado separadamente*) e, após, descrição estrutural para interconectá-los no módulo Multiplicador (vide observação 1). Para facilitar, abaixo é encontrado o diagrama de estados (fig. 3) e de blocos do circuito (fig. 4).

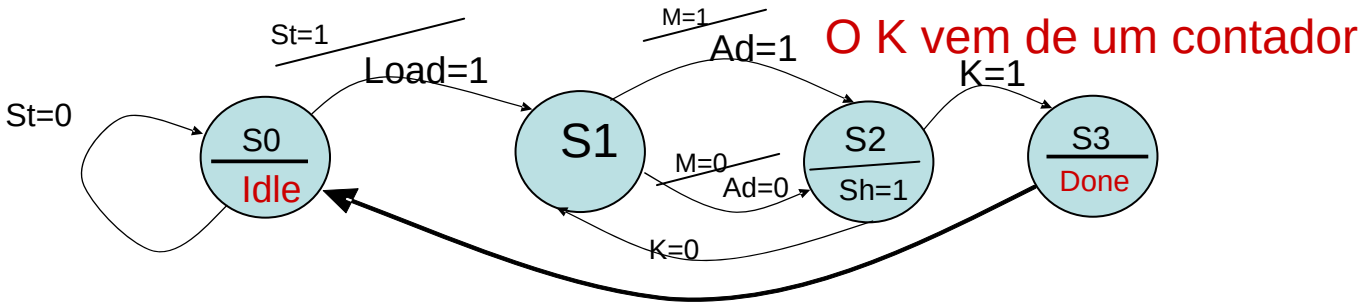

Figura 3: Diagrama de Estados

Legenda:

- *Load* → Carrega os bits 3 a 0 do ACC com o operando *multiplicador* e os bits 8 a 4 do ACC com 0;
- *Sh* (shift) → desloca os bits registrados em ACC 1 bit para a direita;
- *Ad* (add) → Carrega a saída do 4-Bit Adder (resultado da soma) nos bits 8 a 4 do ACC. Não altera o valor dos bits 3 a 0 do ACC;
- St (start) → Sinal para se iniciar a operação;
- K → flag que sinaliza que as ***n*** operações de shift e add necessárias para a multiplicação foram efetuadas (***n*** dependerá do tamanho dos operandos);
- Done → Sinal de indicação de operação finalizada.
- Idle → Sinaliza que o circuito está ocioso.

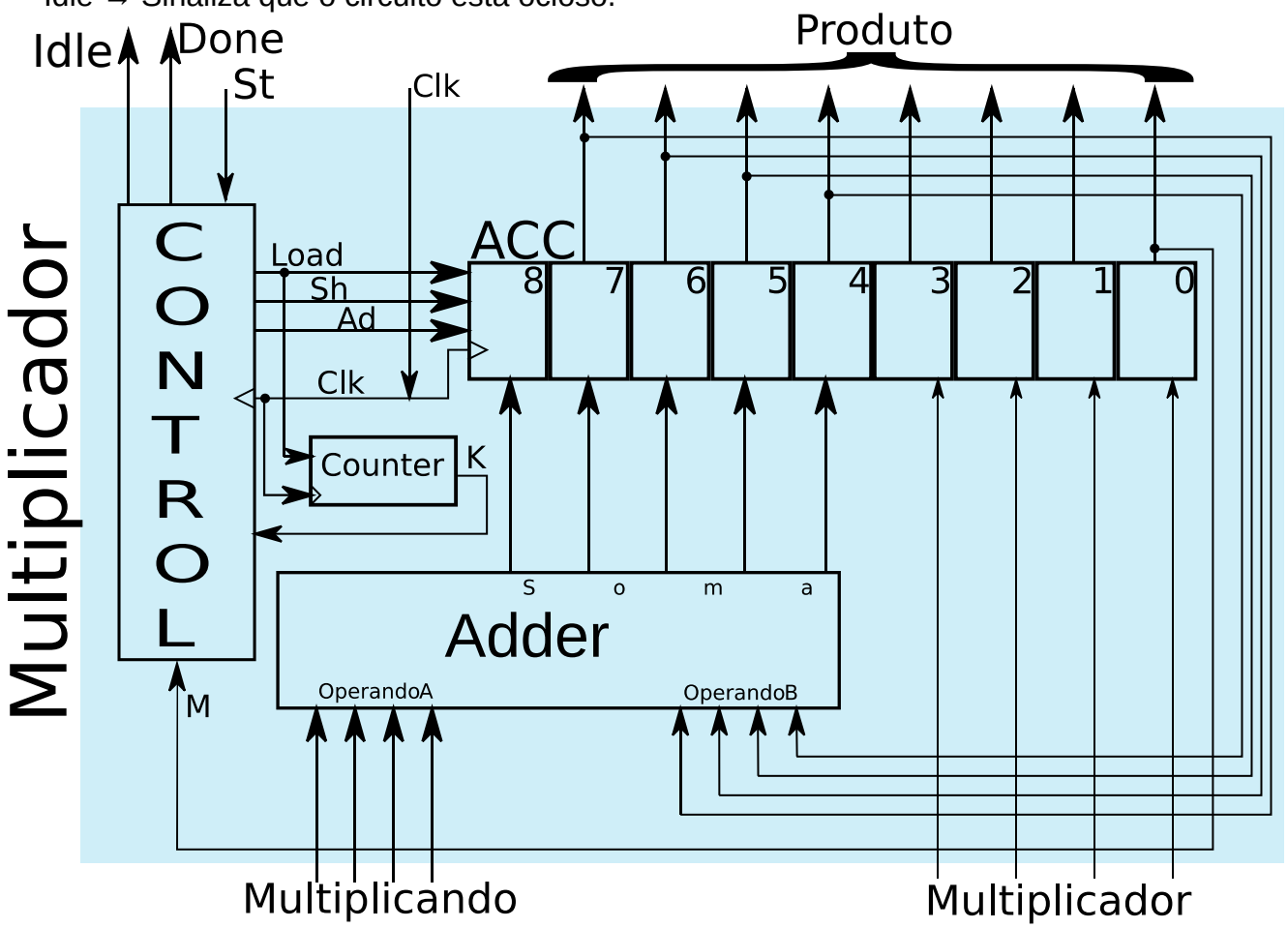

Figura 4: Diagrama de Blocos

Obs.: Se achar necessário, adicionar uma linha de Reset.

3- Implemente, em verilog, um testbench para o teste do circuito Multiplicador e para cada um dos blocos que o compõe.

Observação 1:

Salve a hierarquia de projeto de cada módulo em

- **Alunos\Documentos\Multiplicador\NomeMódulo** (caso esteja sendo utilizado Windows)
- **/home/aluno/Work/Quartus/Multiplicador/NomeMódulo** (caso esteja sendo utilizado Linux)

Já o módulo TOP, **Multiplicador**, será salvo em:

- **Alunos\Documentos\Multiplicador** (caso esteja sendo utilizado Windows)
- **/home/aluno/Work/Quartus/Multiplicador** (caso esteja sendo utilizado Linux)

Lembre-se de que, o módulo *TOP*, Multiplicador, deverá ter em seu projeto os arquivos de implementação de cada módulo que o compõe inseridos através do Quartus (***Menu Project → Add/Remove files in project***).
Dessa forma o projeto fica bem organizado pois cada módulo terá seu próprio projeto associado, podendo ser testado individualmente e utilizado em outros módulos posteriormente.

Entregar a tarefa através do SIGAA.
Arquivo Zipado contendo (delete todos os outros arquivos, pastas e subpastas, a não ser os seguintes arquivos e pastas):

- **Multiplicador**
  - Multiplicador.qpf
  - Multiplicador.qsf
  - Multiplicador.v
  - Multiplicador_TB.v
  - **Adder**
    - Adder.qpf
    - Adder.qsf
    - Adder.v
    - Adder_TB.v
  - **CONTROL**
    - CONTROL.qpf
    - CONTROL.qsf
    - CONTROL.v
    - CONTROL_TB.v
  - **Counter**
    - Counter.qpf
    - Counter.qsf
    - Counter.v
    - Counter_TB.v
  - **ACC**
    - ACC.qpf
    - ACC.qsf
    - ACC.v
    - ACC_TB.v