



**UNIVERSIDADE FEDERAL DE PELOTAS  
CENTRO DE ENGENHARIAS  
CURSOS DE ENGENHARIA ELETRÔNICA  
DISCIPLINA SISTEMAS DIGITAIS I**

**Desenvolvimento de um projeto de  
sistema digital em linguagem VHDL**

**FELIPE G DE LEON**

**TURMA M1 - 1º SEMESTRE 2017**

# Distribuição dos Projetos por aluno

15102796	FELIPE GARCIA DE LEON	1
----------	-----------------------	---

A segunda avaliação do semestre consiste no desenvolvimento de um projeto de sistema digital em linguagem VHDL, e a respectiva validação do mesmo em software. Os projetos devem ser desenvolvidos segundo a metodologia RTL, baseada em parte de controle (máquina de estados) e parte operativa. Além dos códigos VHDL, uma representação gráfica da parte de controle e da parte operativa do sistema deve ser enviada. A avaliação será realizada através da análise da funcionalidade do projeto resultante. Aspectos como clareza e organização dos códigos serão avaliados. O desenvolvimento do projeto durante o horário das aulas também será considerado na avaliação.

O trabalho é individual (cópias terão a nota reduzida). A data para entrega do projeto é 20/08/2016. Atrasos na entrega não serão permitidos, resultando em nota zero na avaliação.

## Projetos:

1. Projeto de uma arquitetura MAC (somador/acumulador) de 16 bits. Fazer duas implementações desta arquitetura: a) uma estrutura com e sem pipeline; b) uma estrutura utilizando memórias RAM e ROM, multiplexador e contador. Realizar uma comparação de desempenho entre elas.

Para descrever o projeto RTL será utilizado o método geral da tabela 5.1 do livro do Vahid o mesmo estuda em sala de aula:

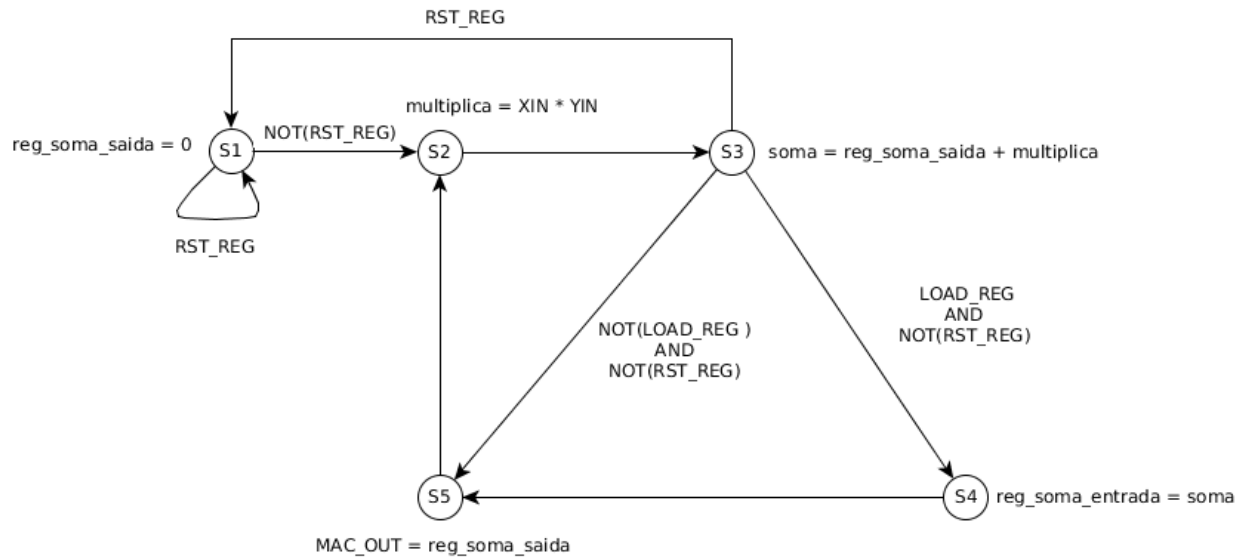
**TABELA 5.1** Método de projeto RTL

	<b>Passo</b>	<b>Descrição</b>
Passo 1	Obtenha uma máquina de estados de alto nível	Descreva o comportamento desejado do sistema na forma de uma máquina de estados de alto nível. Essa máquina consiste em estados e transições. A máquina de estados é de "alto nível" porque as condições para as transições e as ações dos estados são mais do que simplesmente operações booleanas envolvendo os bits de entrada e de saída.
Passo 2	Crie um bloco operacional	Partindo da máquina de estados de alto nível do passo anterior, crie um bloco operacional capaz de realizar as operações que envolvem dados.
Passo 3	Conecte o bloco operacional a um bloco de controle	Conecte o bloco operacional a um bloco de controle. Conecte também as entradas e saídas booleanas que são externas ao bloco de controle.
Passo 4	Obtenha a FSM do bloco de controle	Converta a máquina de estados de alto nível na máquina de estados finitos do bloco de controle (FSM). Para isso, substitua as operações que envolvem dados por sinais de controle, que são ativados ou lidos pelo bloco de controle.

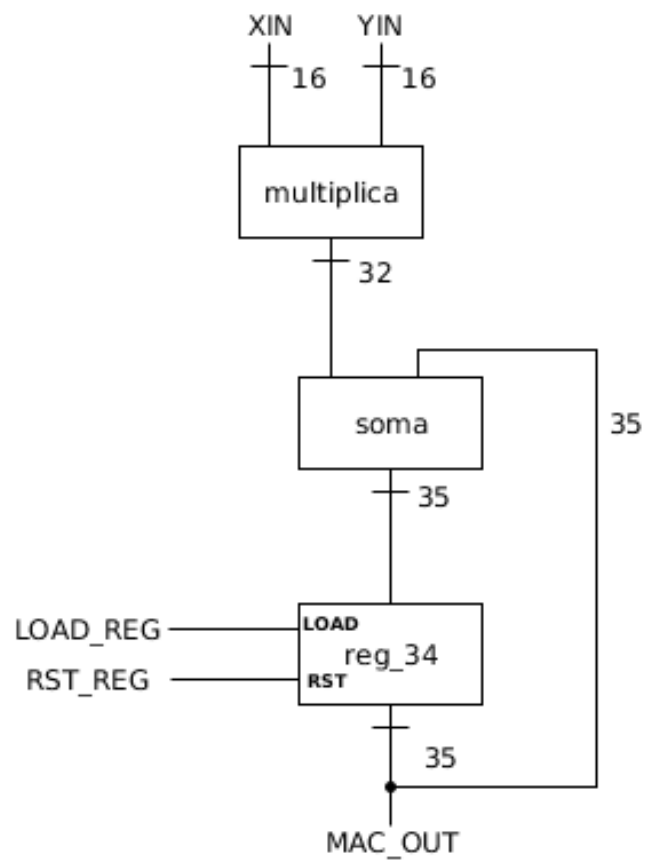
Apos a apresentação dos diagramas de cada etapa será feita uma breve descrição técnica do projeto e a apresentação do funcionamento através dos gráficos dos testbench.

## a) Estrutura MAC sem pipeline

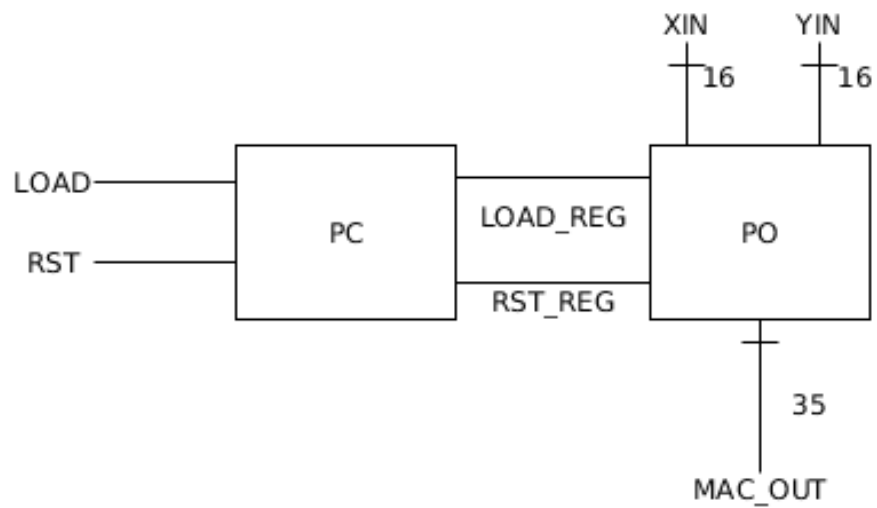
### 1) Diagrama máquina de estados alto nível MAC



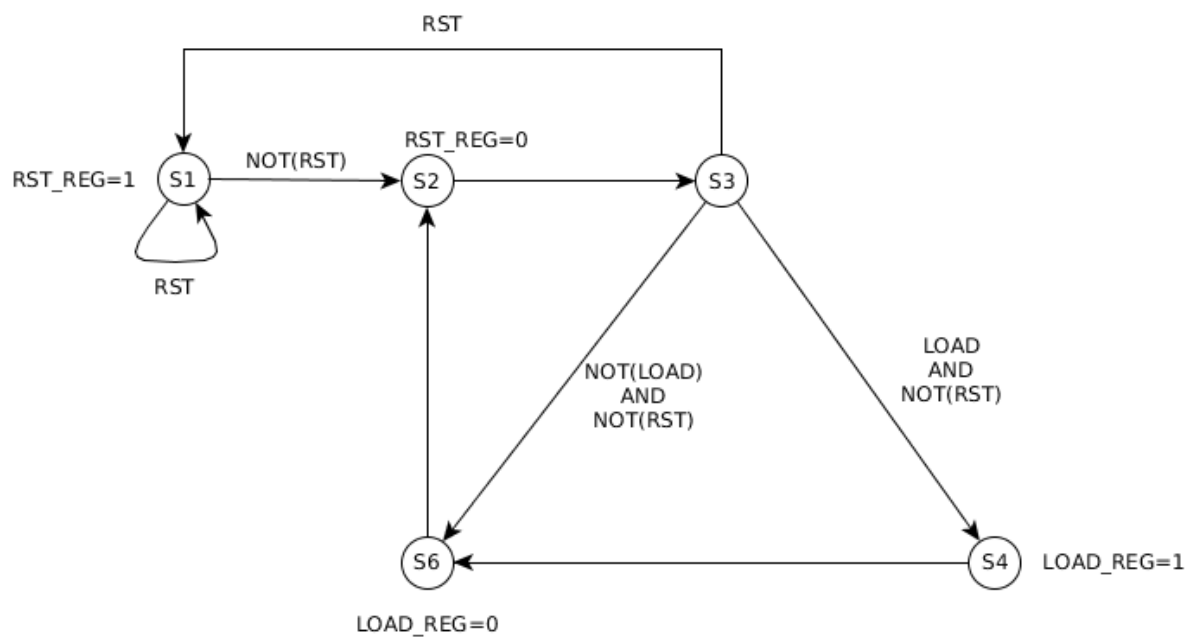
### 2) Diagrama do bloco operacional MAC



### 3) Diagrama de blocos MAC conexão PC-PO



### 4) Diagrama FSM bloco de controle MAC



## MAC, multiplicador somador sem pipeline descrição:

O objetivo deste circuito é demonstrar a funcionalidade básica de um MAC, utilizando sincronismo assíncrono.

O funcionamento básico, quando a entrada **RST** zera o registrador “resetando” a contagem da saída do MAC para ‘0’, o bloco multiplicador tem um tempo de execução de 20  $\mu$ s, o bloco somador 10  $\mu$ s e o registrador 0.2  $\mu$ s, com um tempo total de execução por instrução de 30.2  $\mu$ s, as entradas XIN e YIN são utilizadas pelo multiplicador, a entrada LOAD funciona como uma chave de execução que quando ativa o resultado dos cálculos é apresentado na saída do circuito a cada ciclo de 30.2  $\mu$ s, no momento que este sinal de entrada é desativado o circuito não acumula valores e mostra o último resultado obtido em sua saída.

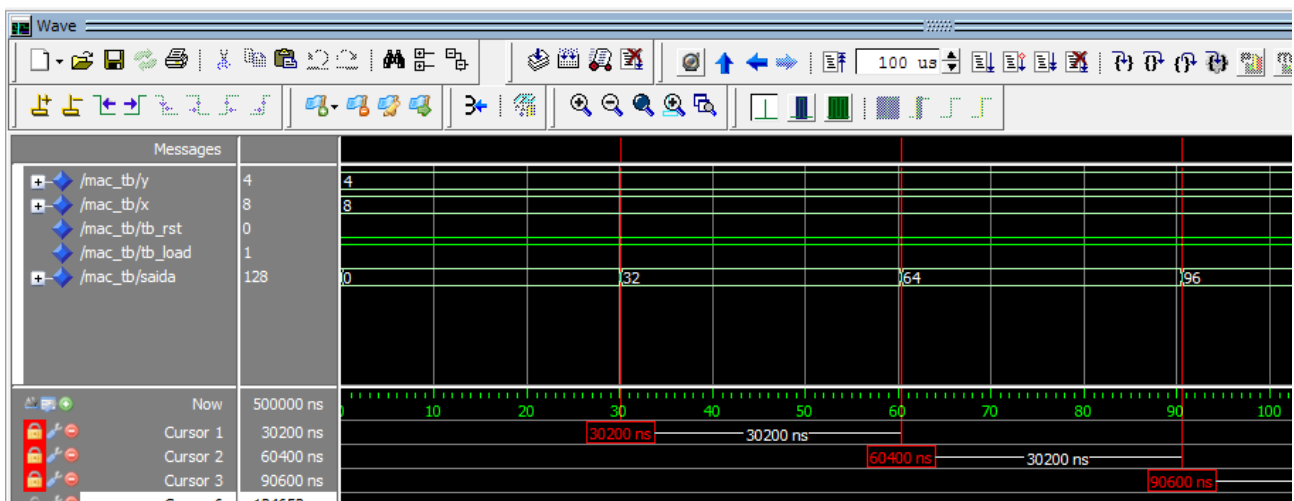
Abaixo a demonstração gráfica dos testes do código vhdl para o MAC sem pipeline:

### Gráfico #1 MAC sem pipeline Funcionamento básico

Primeiro gráfico é para demonstrar o funcionamento básico do circuito.

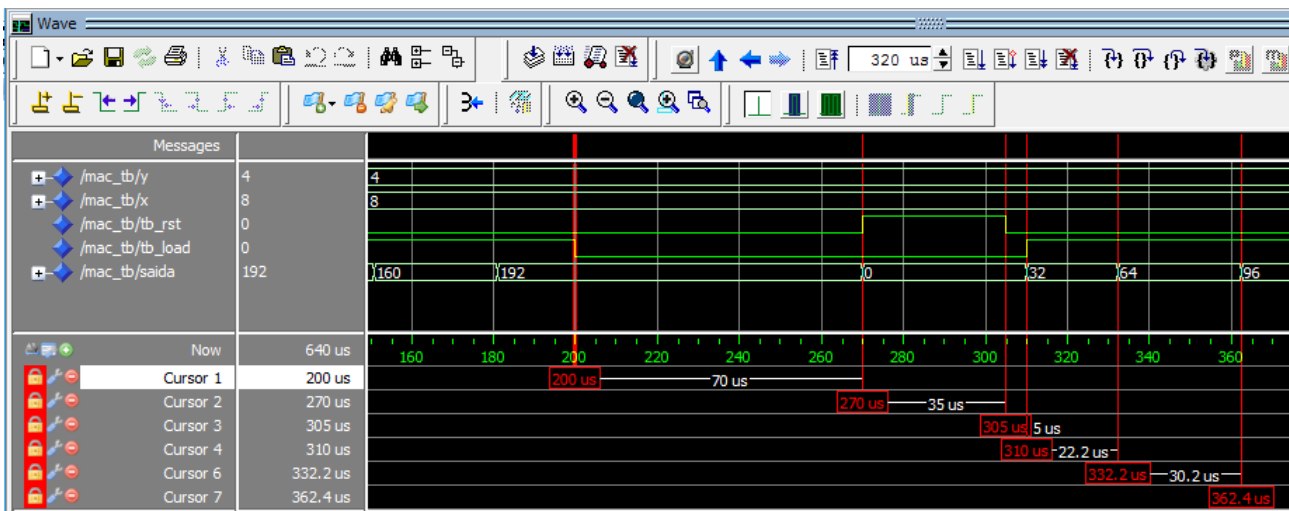
Com o circuito ativo, LOAD em ‘1’ e RST em ‘0’ a saída do circuito apresenta a cada pulso de 30.2  $\mu$ s o valor de uma nova acumulação.

Para facilitar a visualização da execução da acumulação a cada pulso de 30.2  $\mu$ s, cursores foram adicionados a cada 30.2  $\mu$ s.



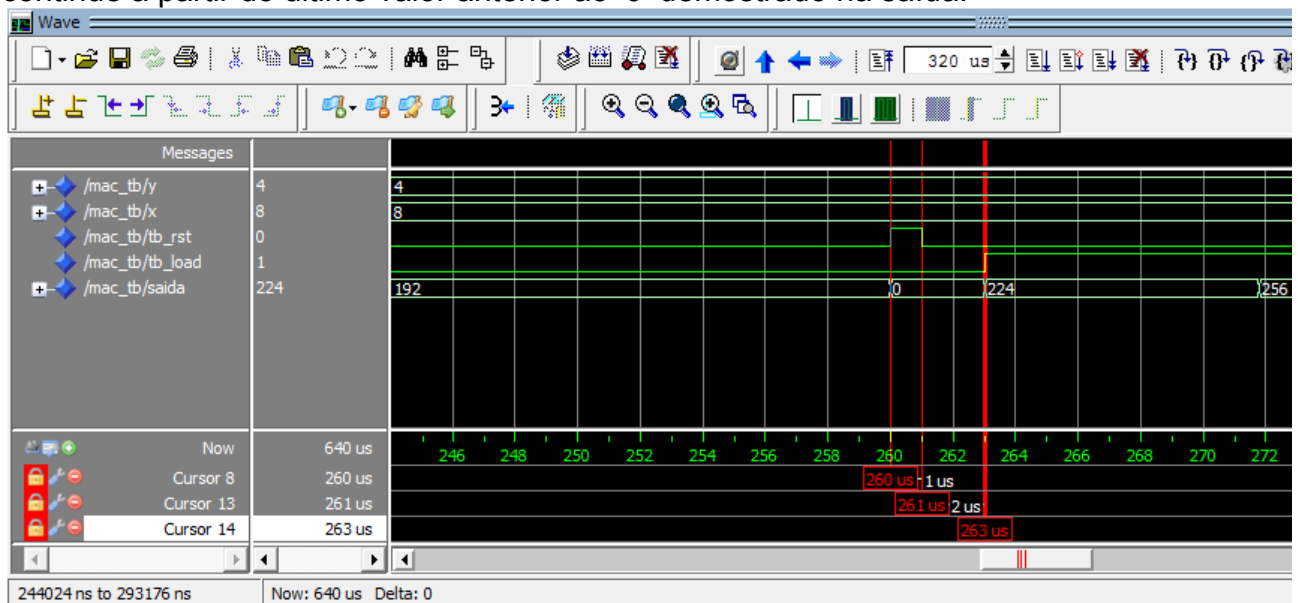
### Gráfico #2 MAC sem pipeline Funcionamento LOAD e RST

Esta segunda imagem é para demonstrar o funcionamento das entradas RST e LOAD, com o LOAD em '0' em 200  $\mu$ s a acumulação para e demonstra um valor constante em "192", em 270  $\mu$ s quando RST passa de '0' para '1' o resultado da acumulação zera, o valor permanece assim enquanto o valor de RST e LOAD não muda, após a entrada RST passar para '0' e a entrada LOAD voltar a ser '1' em 310  $\mu$ s o resultado da primeira acumulação aparece na saída posteriormente após 22.2  $\mu$ s o circuito volta a acumular, este primeiro passo demonstra que mesmo sem o resultado mudar na saída o bloco somador e multiplicador estão executando sempre que o circuito está ativo, posteriormente a contagem volta a acumular a cada pulso de 30.2  $\mu$ s como esperado e assim segue caso não tenha mudança nas entradas.



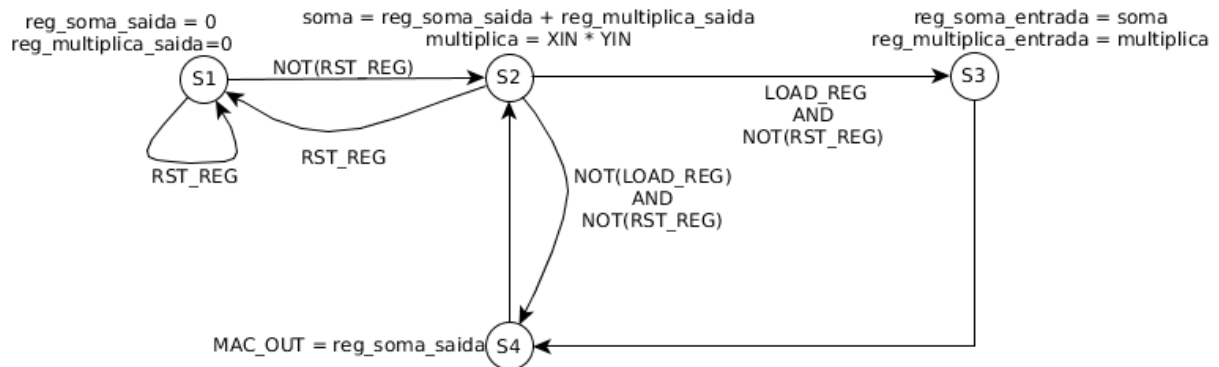
### Gráfico #3 MAC sem pipeline, análise da limitação do circuito

Devido este circuito ser assíncrono e simples em termos de funcionalidade e construção demonstrou ter um limite, sua entrada RST tem de permanecer ativa em '1' por um tempo maior que um ciclo de  $30.2 \mu s$  caso isso não ocorra é possível que o valor de entrada da soma que vem do registrador não seja percebido pelo bloco de soma e a contagem continue a partir do último valor anterior ao '0' demonstrado na saída.

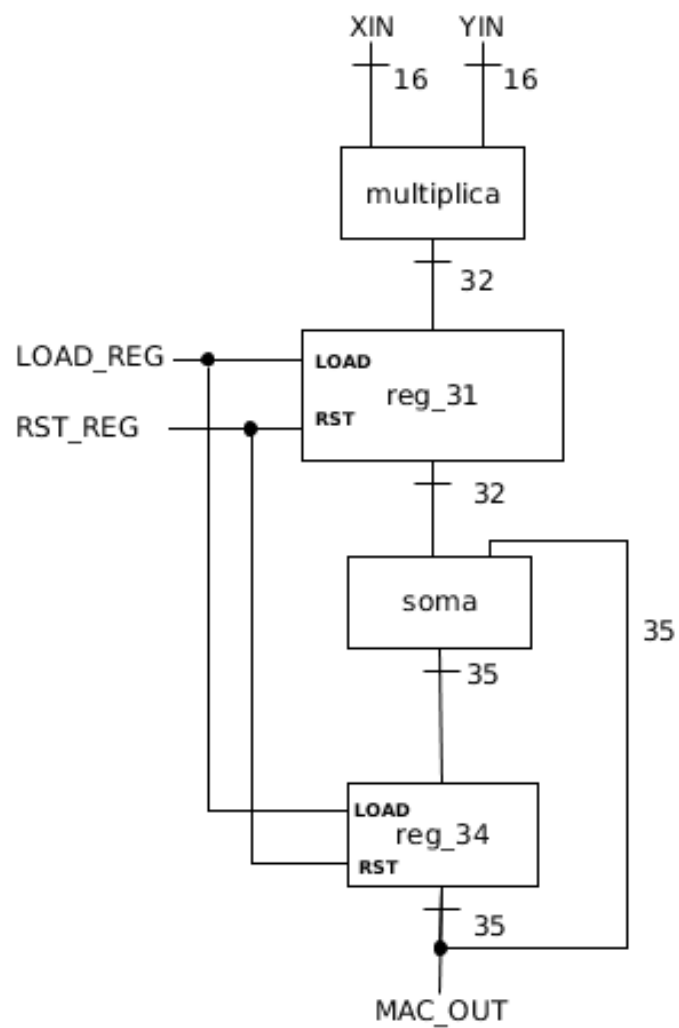


## a) Estrutura MAC com pipeline

### 1) Diagrama máquina de estados alto nível MAC com pipeline

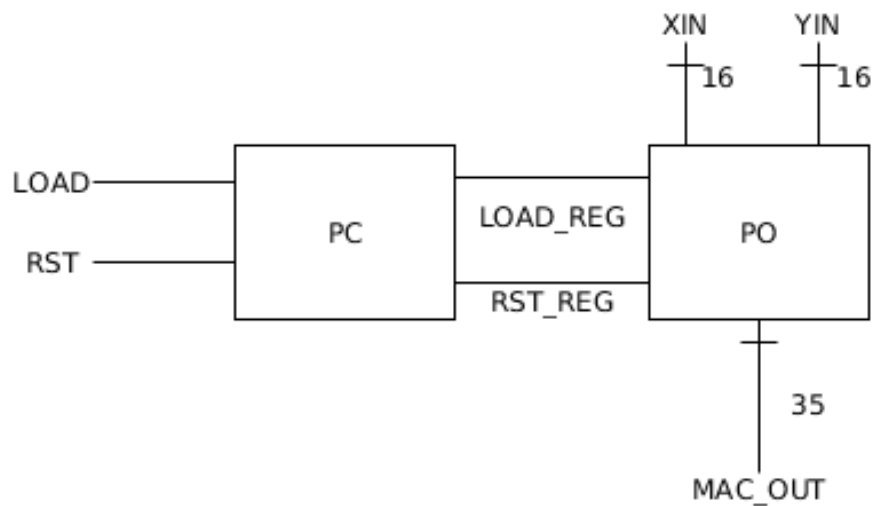


### 2) Diagrama do bloco operacional MAC com pipeline



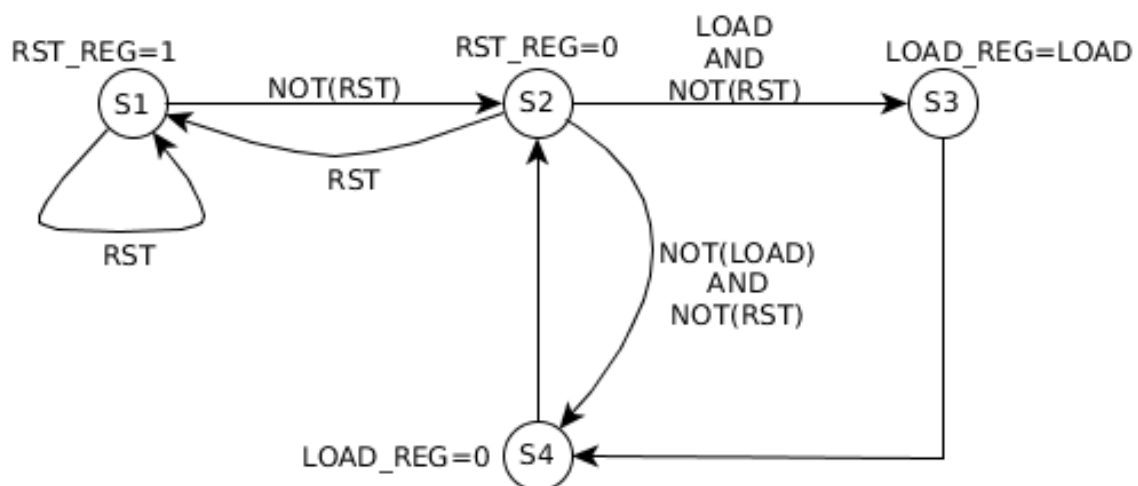


### 3) Diagrama de blocos MAC com pipeline conexão PC-PO



Obs.: em relação ao diagrama do MAC sem pipeline não existe diferença em relação ao com pipeline pois os registradores estão sincronizados em relação ao sinal de LOAD\_REG e RST\_REG.

### 4) Diagrama FSM bloco de controle MAC com pipeline



## MAC, multiplicador somador com pipeline descrição:

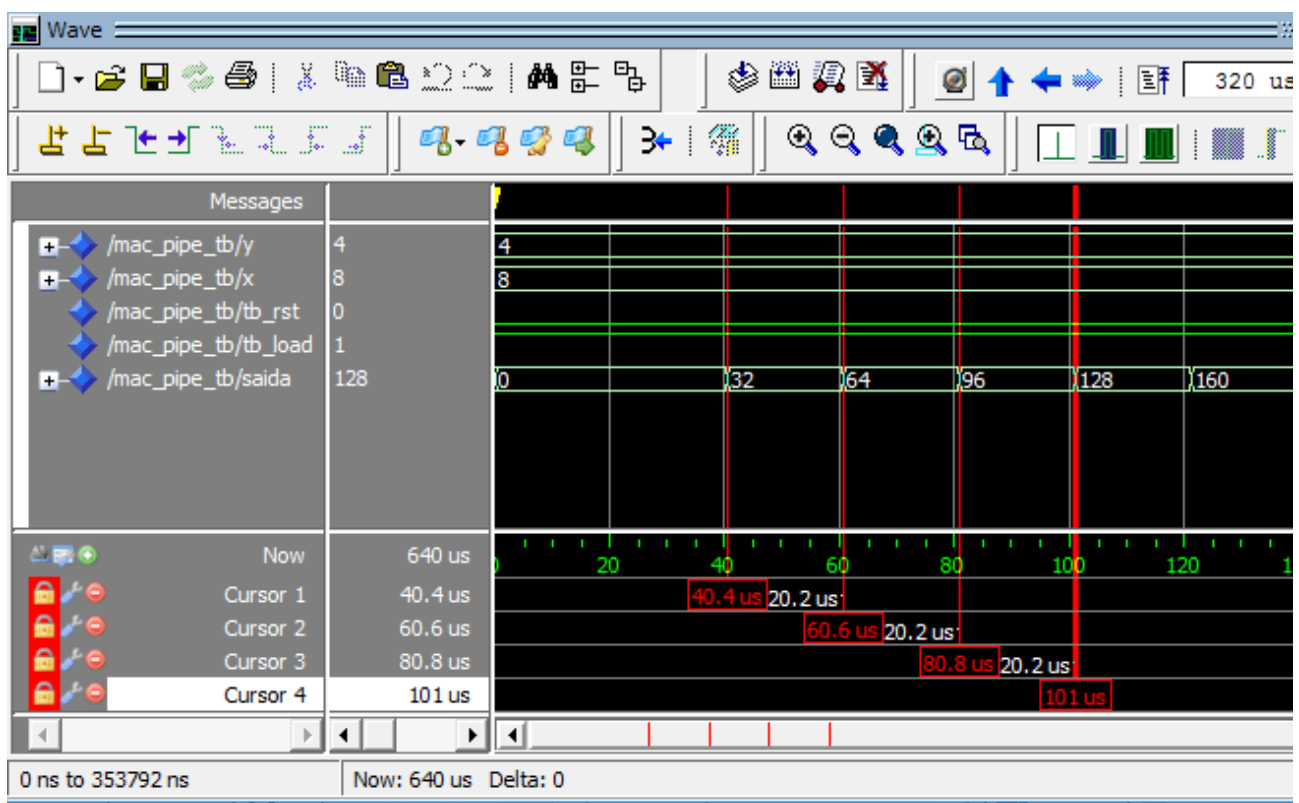
O objetivo deste circuito é demonstrar a funcionalidade básica de um MAC, utilizando sincronismo assíncrono e um nível de pipeline.

O funcionamento básico, quando a entrada **RST** zera o registrador “resetando” a contagem da saída do MAC para ‘0’, o bloco multiplicador e o somador tem um tempo de execução simultâneo de 20  $\mu$ s e os registradores 0.2  $\mu$ s simultâneos, com um tempo total de execução por instrução de 20.2  $\mu$ s (após a latência inicial da primeira contagem), as entradas XIN e YIN são utilizadas pelo multiplicador, a entrada LOAD funciona como uma chave de execução que quando ativa o resultado dos cálculos é apresentado na saída dos registradores e por consequência na saída do circuito, no momento que este sinal de entrada esta desativado o circuito não acumula valores e mostra o ultimo resultada obtido em sua saída.

Abaixo a demonstração gráfica do teste do código vhdl para o MAC com pipeline:

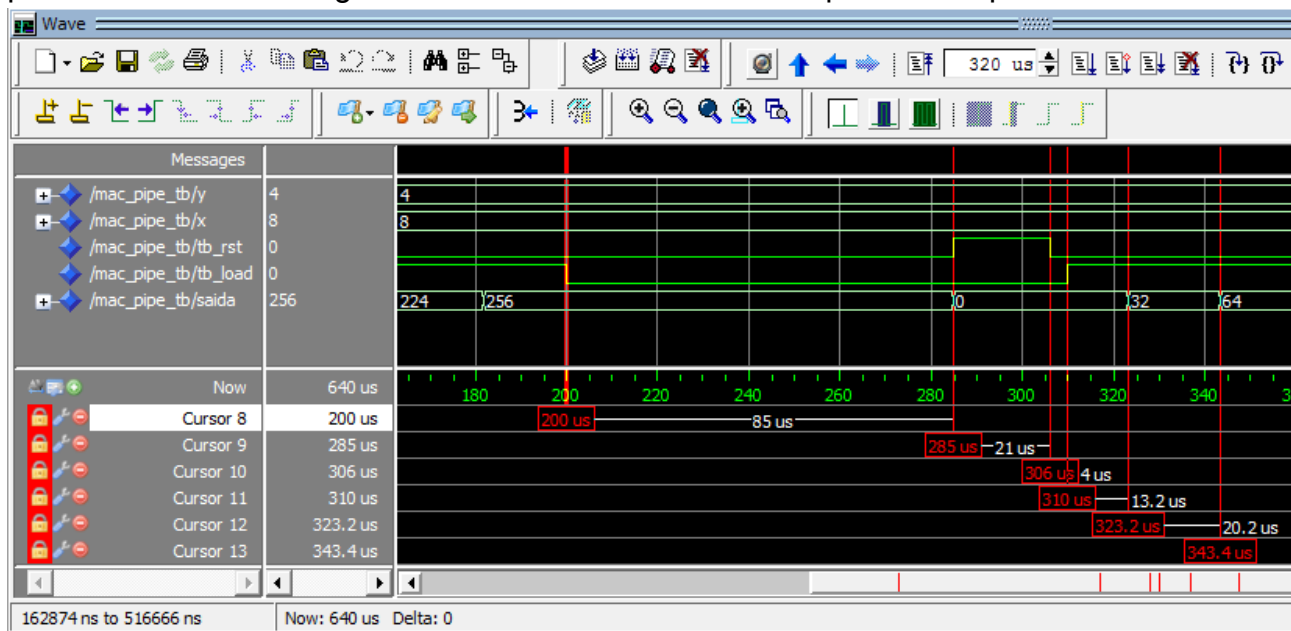
### Gráfico #1 MAC com pipeline Funcionamento básico

Exatamente como estudado na teoria, o primeiro resultado só é computado após um ciclo total de 40.4  $\mu$ s que é a soma do tempo de execução de uma soma mais uma multiplicação e dois registradores, após o primeiro resultado ser afetado pela latência inicial, os resultados seguintes apresentarão um ciclo de 20.2  $\mu$ s exatamente como o esperado, cursores foram adicionados nas mudanças pra visualizar facilmente o valor de cada passo.



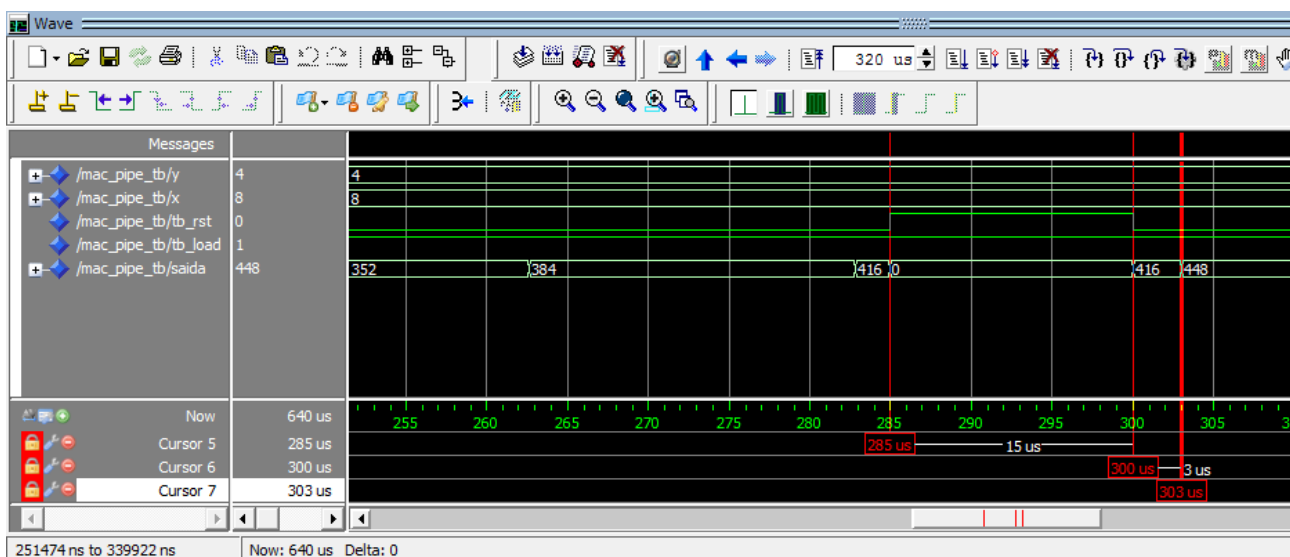
## Gráfico #2 MAC com pipeline Funcionamento LOAD e RST

Esta segunda imagem é para demonstrar o funcionamento das entradas RST e LOAD, quando o LOAD é '0' em 200  $\mu$ s a acumulação para e fica constante em "256", em 285  $\mu$ s quando RST é '1' o resultado da acumulação zera, o valor permanece assim enquanto o valor de RST e LOAD não muda, após a entrada RST zerar e a entrada LOAD ir para '1' em 310  $\mu$ s o resultado da primeira acumulação aparece na saída posteriormente após 13.2  $\mu$ s o circuito volta a acumular, este primeiro passo demonstra que mesmo sem o resultado aparecer na saída o bloco somador e multiplicador estão executando, posteriormente a contagem volta a acumular a cada 20.2  $\mu$ s como esperado.



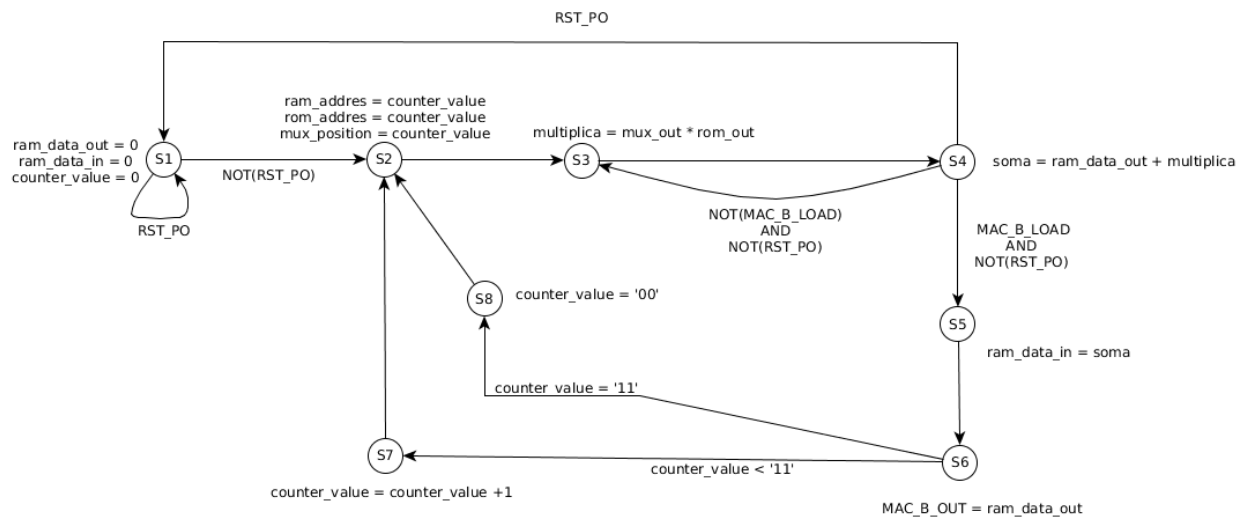
## Gráfico #3 MAC com pipeline, análise da limitação do circuito:

Devido este circuito ser assíncrono e simples em termos de funcionalidade e construção, demonstrou ter um limite, usa entrada RST tem de permanecer em '1' por um tempo maior que um ciclo de 20.2  $\mu$ s caso isso não ocorra é possível que o valor da soma não resete e a contagem continue a partir do último valor anterior ao '0' da saída.

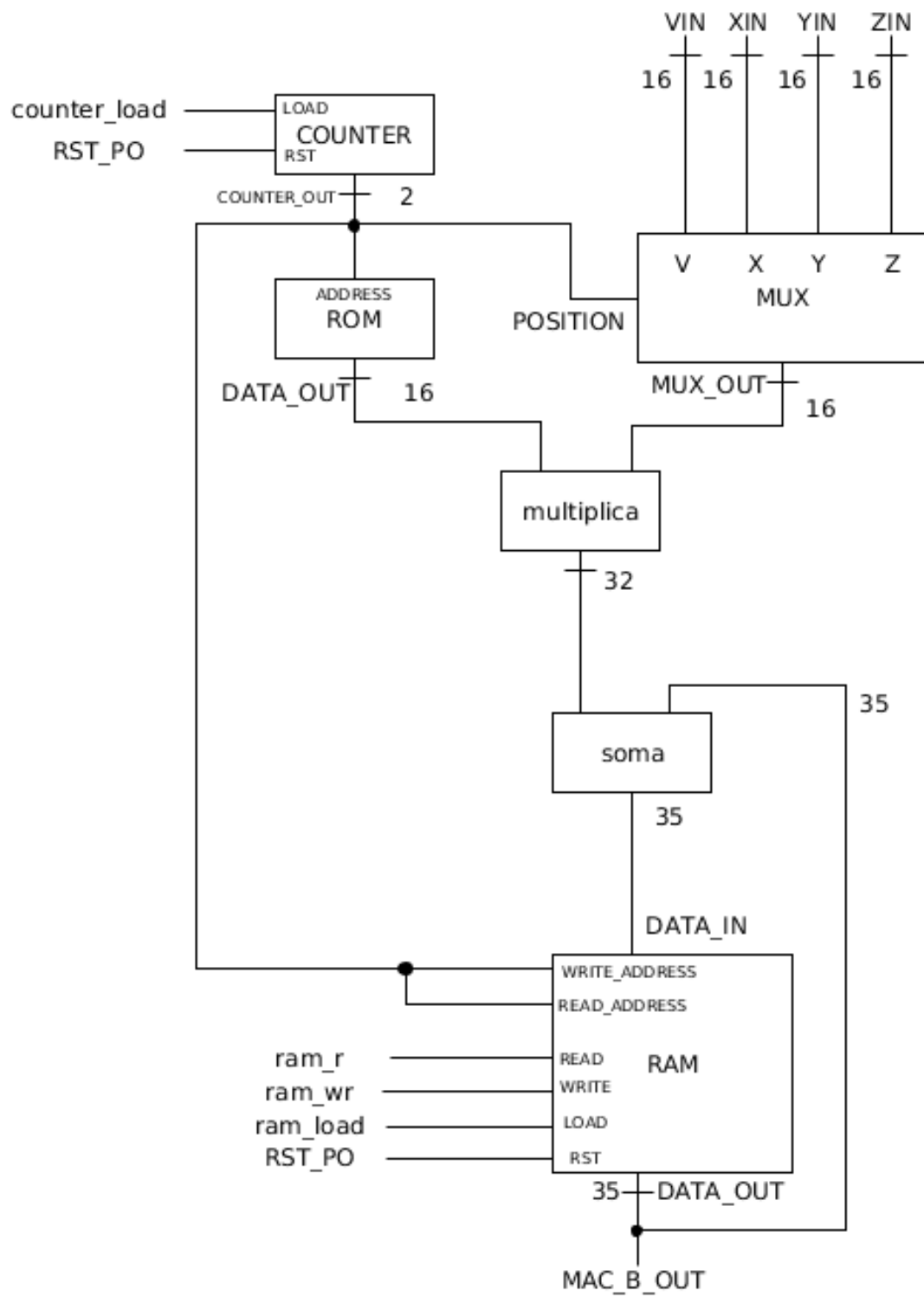


**b) Estrutura MAC sem pipeline, utilizando memórias RAM e ROM, multiplexador e contador.**

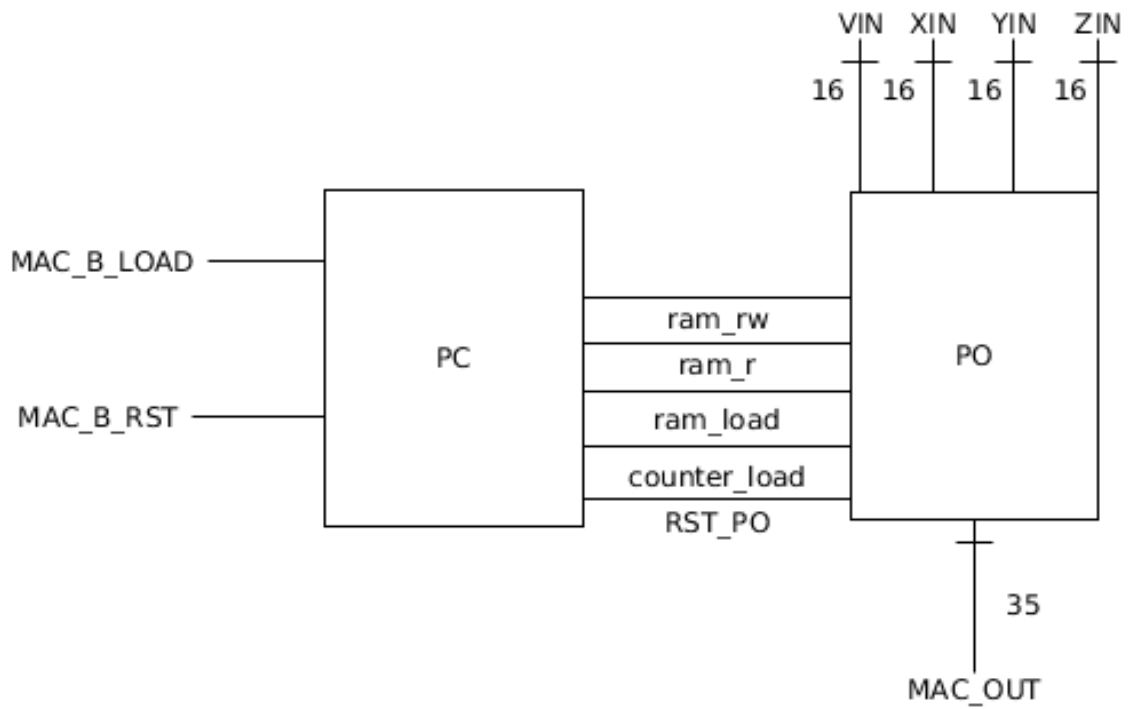
**1) Diagrama máquina de estados alto nível MAC**



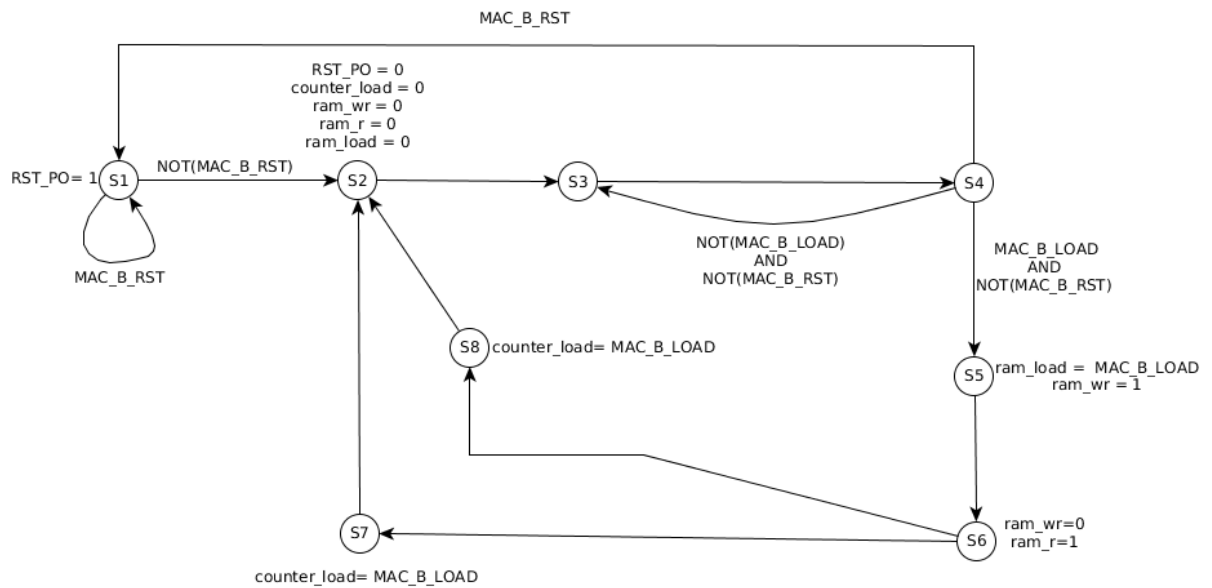
## 2) Diagrama do bloco operacional MAC



### 3) Diagrama de blocos MAC conexão PC-PO



### 4) Diagrama FSM bloco de controle MAC



## MAC sem pipeline, utilizando memórias RAM e ROM, multiplexador e contador.

O objetivo deste circuito é demonstrar a funcionalidade básica de um MAC, utilizando sincronismo assíncrono e memórias RAM e ROM, multiplexador e contador.

O funcionamento básico, quando a entrada **RST** for ativada em '1' todos valores de entrada e saída da **RAM** e do **contador** zera, assim a saída do circuito também zera, o bloco multiplicador tem um tempo de execução de 20  $\mu$ s, o bloco somador 10  $\mu$ s e a RAM 0.2  $\mu$ s, com um tempo total de execução por instrução de 30.2  $\mu$ s mais os tempos de load internos de 0.1 ps que foi utilizado apenas para que o processo funcionasse, o contador neste circuito funciona como um seletor, utilizado para selecionar o canal do MUX e o endereço das memórias, as entradas do MUX são as entradas de valores do sistema, a cada ciclo a multiplicação recebe o valor selecionado do MUX e da ROM, após o calculo finalizar a soma utiliza o valor atual da saída da RAM e soma com o valor atual da multiplicação, este valor é carregado para entrada da RAM, o sistema carrega o valor na RAM ativa o contador para este somar '+1' e o ciclo retorna para a partir da multiplicação em um loop infinito ate que as entradas RST ou LOAD mudem.

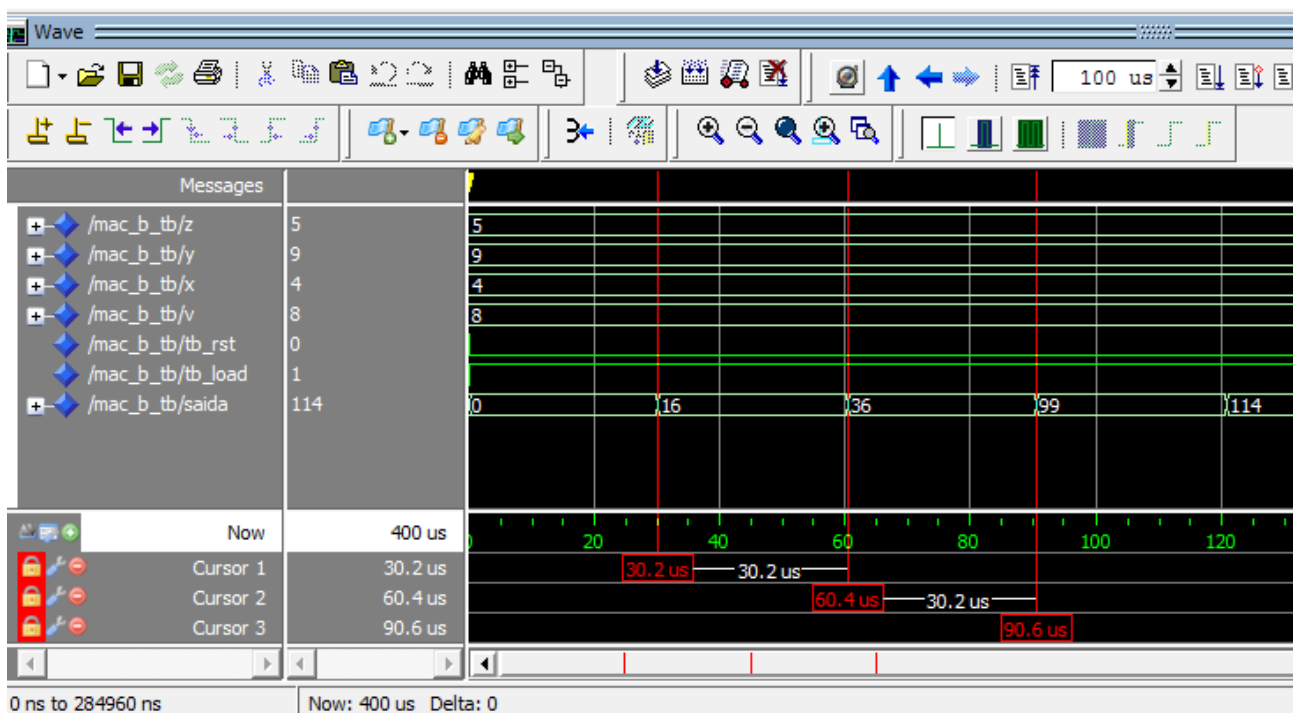
Abaixo a demonstração gráfica dos testes do código vhdl para o MAC sem pipeline:

### Gráfico #1 MAC sem pipeline, utilizando memórias RAM e ROM, multiplexador e contador.

Primeiro gráfico é para demonstrar o funcionamento básico do circuito.

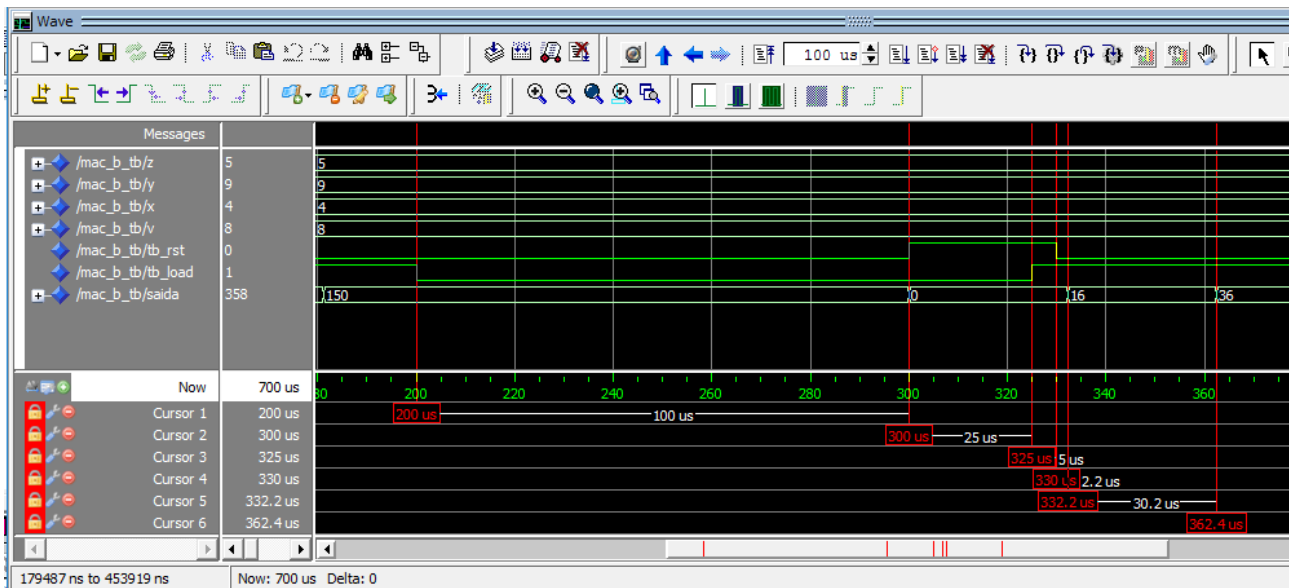
Com o circuito ativo, LOAD em '1' e RST em '0' a saída do circuito apresenta a cada pulso de 30.2  $\mu$ s o valor de uma nova acumulação.

Para facilitar a visualização da execução da acumulação a cada pulso de 30.2  $\mu$ s, cursores foram adicionados a cada 30.2  $\mu$ s.



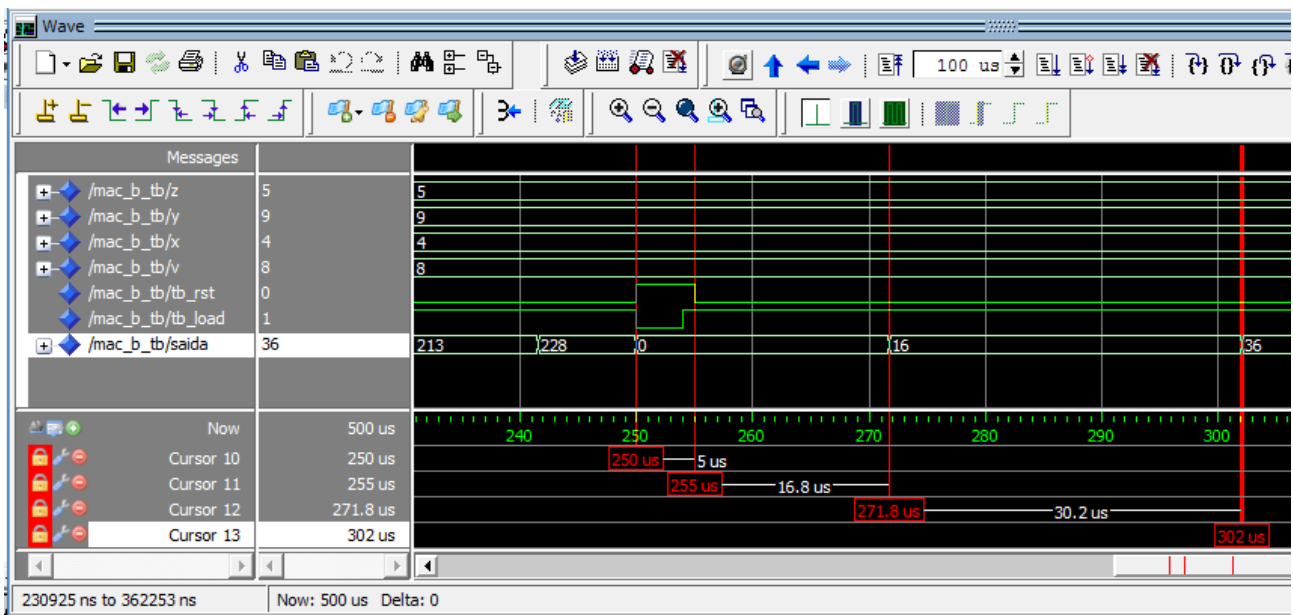
## Gráfico #2 MAC sem pipeline, utilizando memórias RAM e ROM, multiplexador e contador.

Esta segunda imagem é para demonstrar o funcionamento das entradas RST e LOAD, com o LOAD em '0' em 200  $\mu$ s a acumulação para e fica constante em "192", em 300  $\mu$ s quando RST é '1' o resultado da acumulação zera, o valor permanece assim enquanto o valor de RST e LOAD não muda, após a entrada RST zerar e a entrada LOAD voltar a ser '1' em 310  $\mu$ s o resultado da primeira acumulação aparece na saída posteriormente após 2.2  $\mu$ s o circuito volta a acumular, este primeiro passo demonstra que mesmo sem o resultado mudar na saída o bloco somador e multiplicador estão executando, posteriormente a contagem volta a acumular a cada 30.2  $\mu$ s como esperado e assim segue caso não tenha mudança nas entradas.



## Gráfico #3 MAC sem pipeline, utilizando memórias RAM e ROM, multiplexador e contador.

Esta versão do MAC não demonstrou limitação no tempo de duração do pulso de RST como as anteriores. Abaixo foi executado um resete com uma duração de 5  $\mu$ s logo após uma mudança do acumulador, e é possível verificar que isto não causou alteração dos próximos resultados.





## **Comparação de desempenho entre as versões.**

A comparação entre o primeiro MAC sem pipeline e o segundo com pipeline é exatamente o que se esperava em reação a teoria, uma diminuição do tempo de execução por instrução, mostrando que adicionando um registrador e executando simultaneamente dois blocos se obtém um circuito mais eficiente em termos de tempo para se obter as repostas.

A comparação entre o primeiro MAC sem pipeline e como o MAC sem pipeline utilizando memórias RAM e ROM, multiplexador e contador em termos de performance mostrou que não se tem ganhos de desempenho é claro mantendo os valores de tempo nas mesmas escalas para o tempo de registro para o registrador e a RAM ou seja os circuitos operam exatamente da mesma forma.

Caso se utilizasse um pipeline no MAC sem pipeline, utilizando memórias RAM e ROM, multiplexador e contador, o resultado seria o mesmo em termos de tempo do MAC com pipeline.

O que vale ressaltar é que se em termos de desempenho temos o mesmo resultado, em termos de confiabilidade o MAC sem pipeline, utilizando memórias RAM e ROM, multiplexador e contador obteve o melhor resultado pois como é possível verificar pelos gráficos este foi o único que demonstrou controle e resposta absoluta em relação as entradas de RST e LOAD, porém a complexidade é um pouco maior.

Assim é possível concluir que os resultados obtidos nas simulações é satisfatório e de comum acordo com as teorias vistas em sala de aula e encontrada nos livros, também é possível verificar que modificações na escolha dos componentes utilizados pelo circuito e sua configuração pode resultar na melhor eficiência e ou confiabilidade de uso.

**Bibliografia:**

Sistemas Digitais - Projetos de Otimização e HDLs – VAHID (2008)