DESCRIÇÕES DAS 03 INSTRUÇÕES – A seguir é apresentada uma descrição e formatação das instruções.

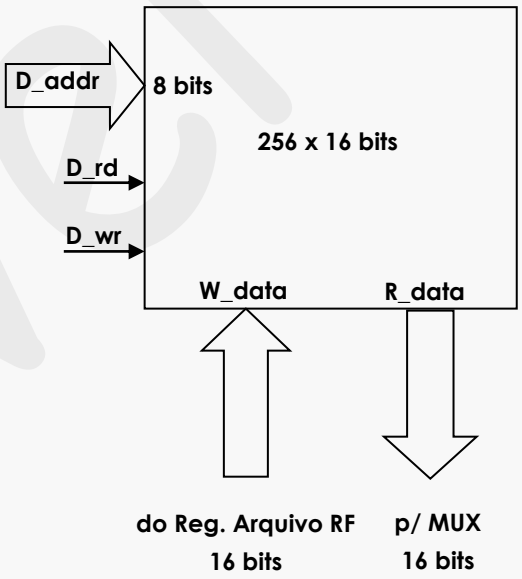
1. Formatação da instrução

Bits	Tipo	Comentário
15...12	Opcde	0000 – Carregar 0001 – Armazenar 0010 – Somar
11...8	ra	0000 – Reg.0 1111 – Reg. 15
7...4	rb	0000 – Reg.0 1111 – Reg. 15
3...0	rc	0000 – Reg.0 1111 – Reg. 15
7...0	d	Posição 0 a 255

2. Instrução Carregar – 0000 $r_3r_2r_1r_0$ $d_7d_6d_5d_4d_3d_2d_1d_0$ – Onde é definido por:

0000 – Opcode da instrução carregar.
$r_3r_2r_1r_0$ – Destino dos dados. $d_7d_6d_5d_4d_3d_2d_1d_0$ – Fonte dos dados.

r_3	r_2	r_1	r_0	Registrador destino
0	0	0	0	Reg. 0
0	0	0	1	Reg.1
0	0	1	0	Reg.2
0	0	1	1	Reg. 3
0	1	0	0	Reg. 4
0	1	0	1	Reg.5
0	1	1	0	Reg.6
0	1	1	1	Reg.7
1	0	0	0	Reg.8
1	0	0	1	Reg. 9
1	0	1	0	Reg. 10
1	0	1	1	Reg. 11
1	1	0	0	Reg.12
1	1	0	1	Reg.13
1	1	1	0	Reg.14
1	1	1	1	Reg.15



3. Instrução Armazenar - 0001 $r_3r_2r_1r_0$ $d_7d_6d_5d_4d_3d_2d_1d_0$ – Onde é definido por:

0001 – Opcode da instrução armazenar.
$r_3r_2r_1r_0$ – Fonte dos dados. $d_7d_6d_5d_4d_3d_2d_1d_0$ – Destino dos dados.

Conforme tabela acima os dados são transferidos do registrador fonte dado por **$r_3r_2r_1r_0$** para a posição de memória dada por **$d_7d_6d_5d_4d_3d_2d_1d_0$**.

4. Instrução Somar – 0010 **$ra_3ra_2ra_1ra_0$ $rb_3rb_2rb_1rb_0$ $rc_3rc_2rc_1rc_0$**. Onde é definido por:

0010 – Opcode da instrução somar.

$ra_3ra_2ra_1ra_0$ – Destino do conteúdo executado pela instrução.

$rb_3rb_2rb_1rb_0$ – Fonte dos dados.

$rc_3rc_2rc_1rc_0$.- Fonte dos dados.

EXEMPLO: Programar o processador para a execução de cálculo da expressão a seguir:
Somar o conteúdo dos endereços de memória a seguir: D[5] = D[5] + D[6] + D[7] e armazenar no endereço 5 de memória. Pede-se o programa realizado na memória de instruções do processador.

0: 0000 0000 00000101 // Carrega o registrador 0 com o conteúdo do endereço 5: RF[0] = D[5].
1: 0000 0001 00000110 // Carrega o registrador 1com o conteúdo do endereço 6: RF[1] = D[6].
2: 0000 0010 00000111 // Carrega o registrador 2 com o conteúdo do endereço 7: RF[2] = D[7].
3: 0010 0000 00000001 // Somar os conteúdos dos registradores 0 e 1 e armazenar no reg.0
RF[0] = RF[0] + RF[1];
4: 0010 000000000010 // Somar os conteúdos dos registrasdores 0 e 2 e armazenar no reg.0
RF[0] = RF[0] + RF[2];
5: 0001 0000 00000101 // Armazenar o conteúdo do reg.0 no endereço de memória 5:
D[5] = RF[0].

UNIDADE DE CONTROLE DO PROCESSADOR DE 03 INSTRUÇÕES

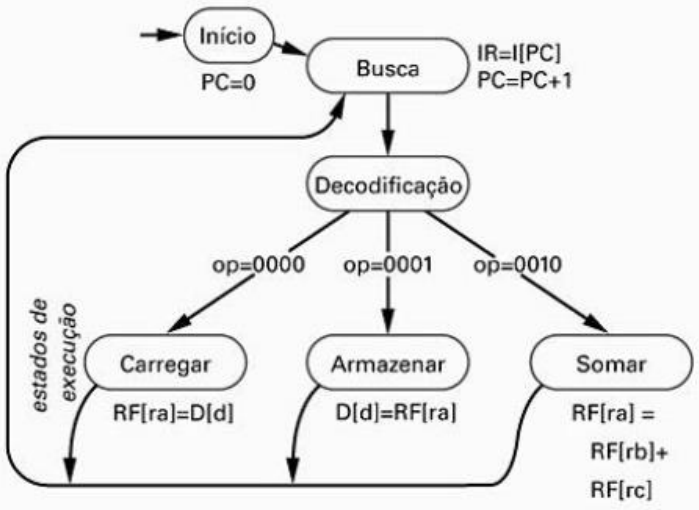


Figura 8.9 Descrição de um processador programável de três instruções por meio de uma máquina de estados de alto nível.

EXEMPLO: DADOS O PROGRAMA DE EXECUÇÃO A SEGUIR:

Linha 0 – RF[0] = D[0]

Linha 1 – RF[1] = D[1]

Linha 2 – RF[2] = RF[0] + RF[1]

Linha 3 – D[9] = RF[2]

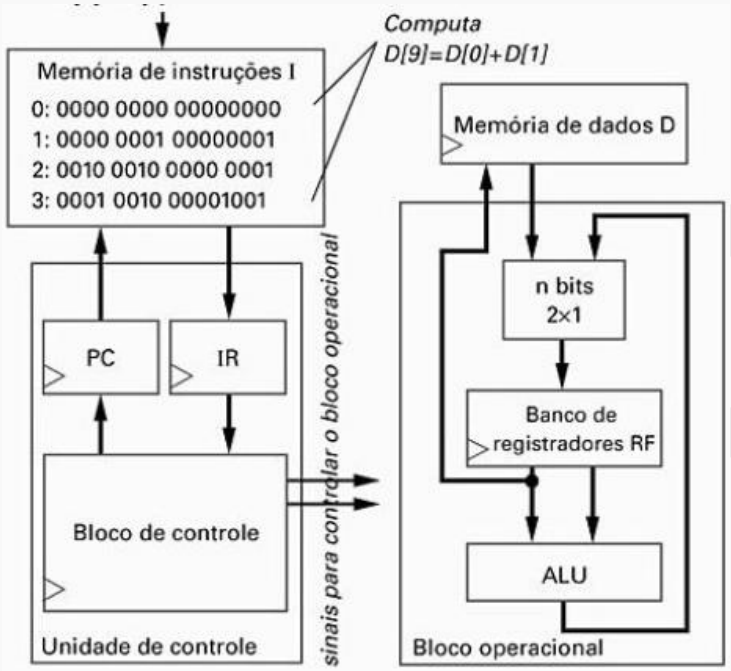


Figura 8.7 Um programa que computa *D[9]=D[0]+D[1]*, usando um dado conjunto de instruções. Inserimos espaços em branco entre os bits da memória de instruções apenas por legibilidade – esses espaços não existem na memória.

F.S.M para o bloco de controle do processador para as 03 instruções.

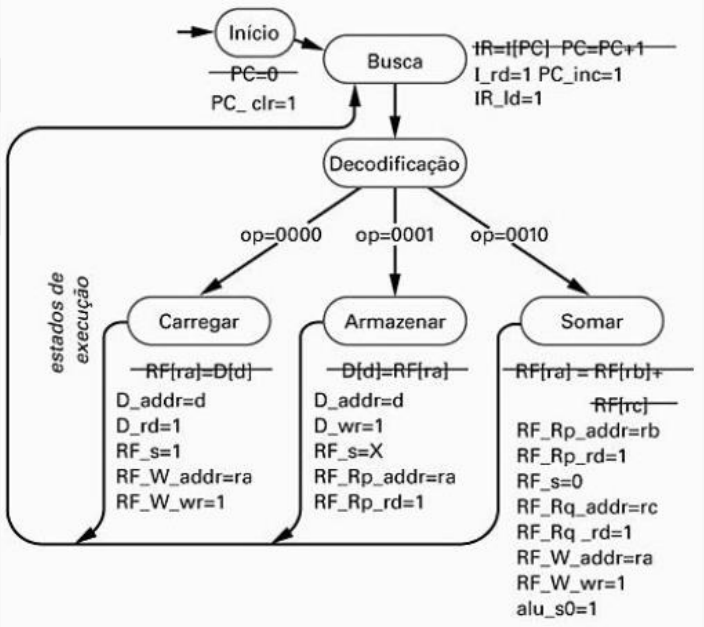
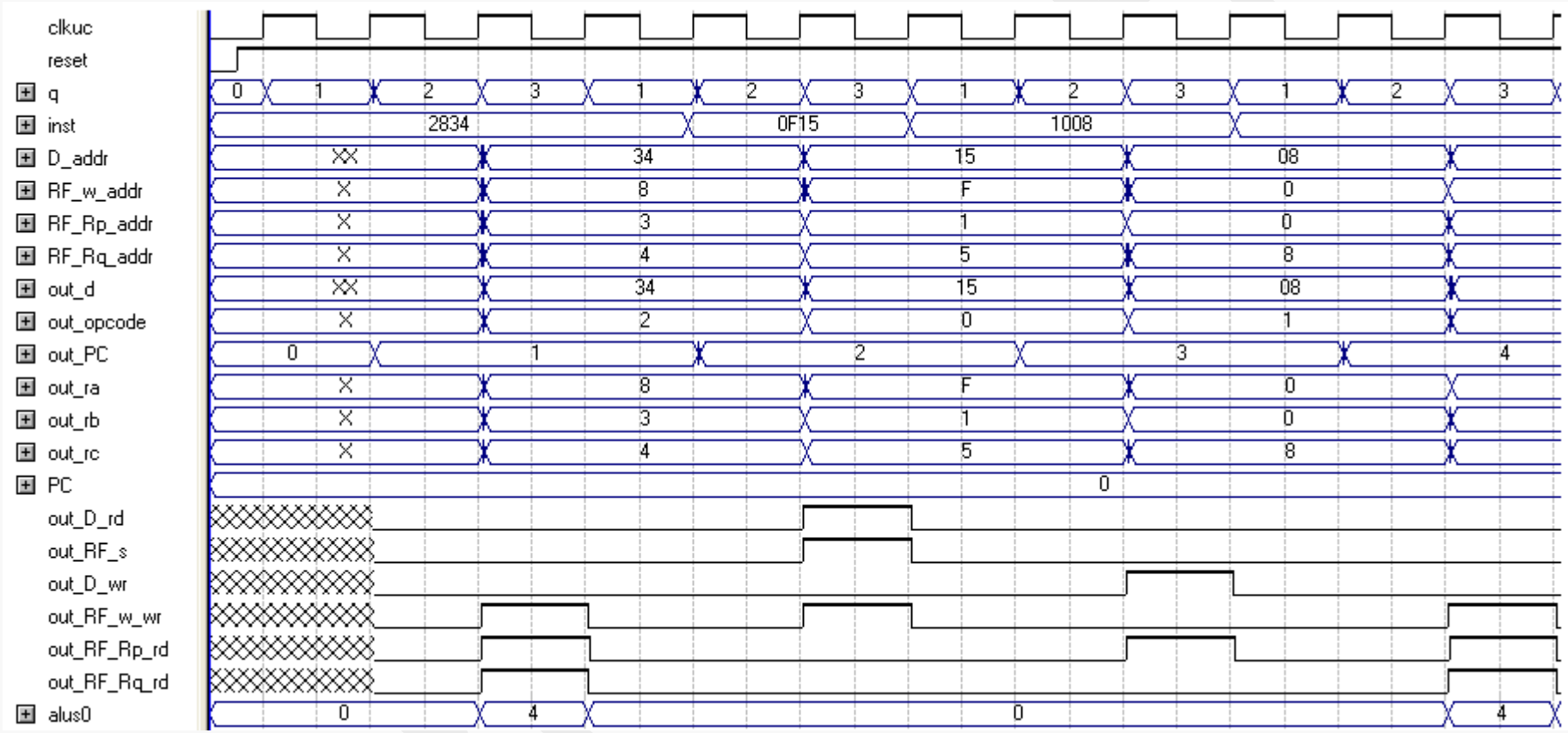
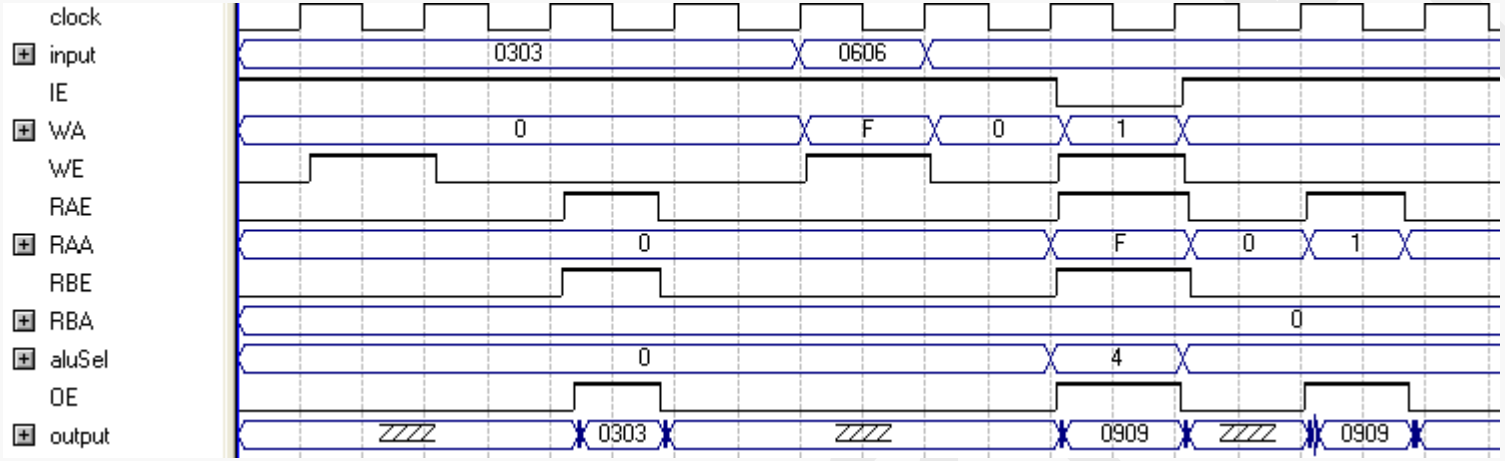


Figura 8.11 FSM para o bloco de controle do processador de três instruções.

Formas de ondas para 03 instruções: Carregar, Armazenar e Somar e a instrução de 16bits sendo bits: opcode(15..12), ra(11..8), rb(7..4), rc(3..0), d(7..0).
Opcode das instruções: 0000 – Carregar, 0001 – Armazenar e 0010 – Somar. Exemplo: Instrução 2834 => opcode 2, ra = 8 e Rb = 3H e rc = 4H, alus0 = 4H.
A instrução pede: RF(8) ← RF(3) + RF(4) e instrução 1005 pede: D(05) = RF(0) e instrução 0F15 pede: RF(F) = D(15).
0F15 => opcode 0,



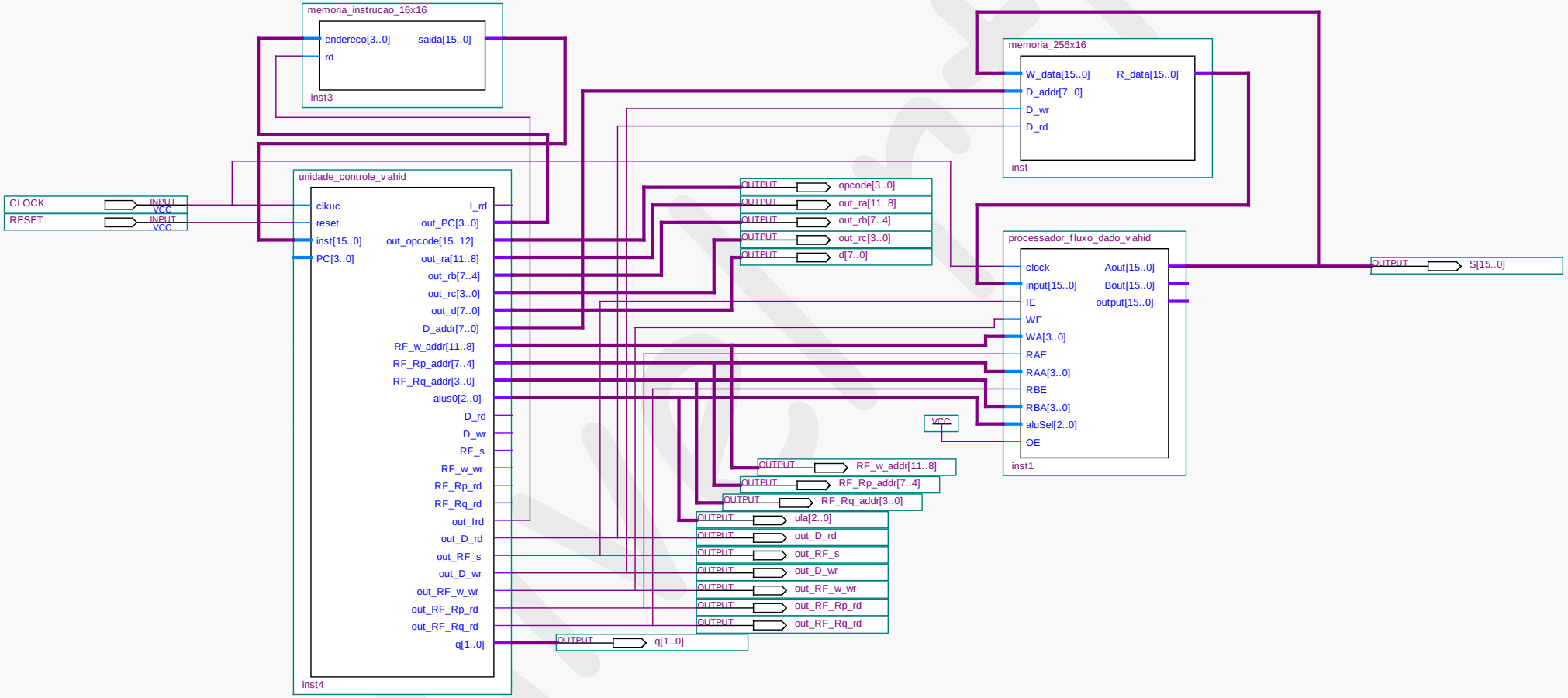
Formas de ondas para o fluxo de dados para duas instruções de carga e uma instrução de soma e instrução de armazenar. A primeira instrução carregar: RF[0} = 0303 e a segunda instrução carregar: RF[15] = 0606. A instrução somar e armazenar : RF[1] = RF[0] + RF[15].
Obs.: Foi introduzido no fluxo de dados uma saída para a simulação das instruções e conferência dos resultados.



Programa a ser executado no processador gera os sinais para o fluxo de dados de acordo com a tabela a seguir.

Linha	instrução	Opcode	Mnem.	D_addr	D_rd	D_wr	RF_s	RF_w_addr	RF_w_wr	RF_Rp_rd	RF_Rp_addr	RF_Rq_rd	RF_Rq_addr	Ula
0	0005	0	RF[0] = D[5]	05	1	0	1	0000	1	0	0000	0	0000	000
1	0106	0	RF[1] = D[6]	06	1	0	1	0001	1	0	0000	0	0000	000
2	0207	0	RF[2] = D[7]	07	1	0	1	0002	1	0	0000	0	0000	000
3	2001	2	RF[0] = RF[0] + RF[1],	xx	0	0	0	0000	1	1	0000	1	0001	100
4	2002	2	RF[0] = RF[0] + RF[2],	xx	0	0	0	0000	1	1	0000	1	0010	100
5	: 1005	1	D[5] = RF[0].	05	0	1	x	0000	0	1	0000	0	0000	000

Circuito do processador para 03 instruções com blocos de controlador unidade de controle, bloco memória de instruções, bloco fluxo de dados e bloco de memória de dados.



O programa da fig. 8.8 é mostrado nas formas de ondas abaixo, onde foi rodado no processador Vahid.