library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.numeric\_std.ALL;

library work;

entity mem is

port(

clk : in std\_logic;

rst\_n : in std\_logic;

out\_mem : out std\_logic\_vector(15 downto 0);

in\_mem : in std\_logic\_vector(15 downto 0);

read : in std\_logic;

write : in std\_logic;

end\_mem : in std\_logic\_vector(15 downto 0);

end mem;

architecture rtl of mem is

subtype word is std\_logic\_vector(7 downto 0);

type mem is array (0 to 255) of word;

signal memoria : mem ;

begin

if(rst\_n = ‘1’) then

memoria(0) <= "00000000"; -- LOAD R0 <= mem(32)

memoria (1) <= "00100000";

memoria (2) <= "00000000"; -- LOAD R1 <= mem(34)

memoria (3) <= "01100010";

memoria (4) <= "00100001"; -- ADD R2 = R0 + R1

memoria (5) <= "10000000";

memoria (6) <= "00010000"; -- STORE R0 => mem(110000)

memoria (7) <= "00110000";

memoria (8) <= "00000000"; -- LOAD R3 <= mem(110000)

memoria (9) <= "11110000";

memoria (10) <= "01010011"; -- R1 = R3 AND R0

memoria (11) <= "01000000";

memoria (12) <= "01110100"; -- JMP zero R1 - R0

memoria (13) <= "00010000"; -- endereco 16

memoria (14) <= "00000000";

memoria (15) <= "00000000";

memoria (16) <= "11110000"; -- HALT

memoria (17) <= "00000000";

memoria (18) <= "00000000";

memoria (19) <= "00000000";

memoria (20) <= "00000000";

memoria (21) <= "00000000";

memoria (22) <= "00000000";

memoria (23) <= "00000000";

memoria (24) <= "00000000";

memoria (25) <= "00000000";

memoria (26) <= "00000000";

memoria (27) <= "00000000";

memoria (28) <= "00000000";

memoria (29) <= "00000000";

memoria (30) <= "00000000";

memoria (31) <= "00000000";

memoria (32) <= "00000000";

memoria (33) <= "00000101";

memoria (34) <= "00000000";

memoria (35) <= "00001000";

memoria (36) <= "00000000";

memoria (37) <= "00000100";

memoria (38) <= "00000000";

memoria (39) <= "00000000";

memoria (40) <= "00000000";

memoria (41) <= "00000000";

memoria (42) <= "00000000";

memoria (43) <= "00000000";

memoria (44) <= "00000000";

memoria (45) <= "00000000";

memoria (46) <= "00000000";

memoria (47) <= "00000000";

memoria (48) <= "00000000";

memoria (49) <= "00000000";

memoria (50) <= "00000000";

memoria (51) <= "00000000";

memoria (52) <= "00000000";

memoria (53) <= "00000000";

memoria (54) <= "00000000";

memoria (55) <= "00000000";

memoria (56) <= "00000000";

memoria (57) <= "00000000";

memoria (58) <= "00000000";

memoria (59) <= "00000000";

memoria (60) <= "00000000";

memoria (61) <= "00000000";

memoria (62) <= "00000000";

memoria (63) <= "00000000";

memoria (64) <= "00000000";

memoria (65) <= "00000000";

memoria (66) <= "00000000";

memoria (67) <= "00000000";

memoria (68) <= "00000000";

memoria (69) <= "00000000";

memoria (70) <= "00000000";

memoria (71) <= "00000000";

memoria (72) <= "00000000";

memoria (73) <= "00000000";

memoria (74) <= "00000000";

memoria (75) <= "00000000";

memoria (76) <= "00000000";

memoria (77) <= "00000000";

memoria (78) <= "00000000";

memoria (79) <= "00000000";

memoria (80) <= "00000000";

memoria (81) <= "00000000";

memoria (82) <= "00000000";

memoria (83) <= "00000000";

memoria (84) <= "00000000";

memoria (85) <= "00000000";

memoria (86) <= "00000000";

memoria (87) <= "00000000";

memoria (88) <= "00000000";

memoria (89) <= "00000000";

memoria (90) <= "00000000";

memoria (91) <= "00000000";

memoria (92) <= "00000000";

memoria (93) <= "00000000";

memoria (94) <= "00000000";

memoria (95) <= "00000000";

memoria (96) <= "00000000";

memoria (97) <= "00000000";

memoria (98) <= "00000000";

memoria (99) <= "00000000";

memoria (100) <= "00000000";

memoria (101) <= "00000000";

memoria (102) <= "00000000";

memoria (103) <= "00000000";

memoria (104) <= "00000000";

memoria (105) <= "00000000";

memoria (106) <= "00000000";

memoria (107) <= "00000000";

memoria (108) <= "00000000";

memoria (109) <= "00000000";

memoria (110) <= "00000000";

memoria (111) <= "00000000";

memoria (112) <= "00000000";

memoria (113) <= "00000000";

memoria (114) <= "00000000";

memoria (115) <= "00000000";

memoria (116) <= "00000000";

memoria (117) <= "00000000";

memoria (118) <= "00000000";

memoria (119) <= "00000000";

memoria (120) <= "00000000";

memoria (121) <= "00000000";

memoria (122) <= "00000000";

memoria (123) <= "00000000";

memoria (124) <= "00000000";

memoria (125) <= "00000000";

memoria (126) <= "00000000";

memoria (127) <= "00000000";

memoria (128) <= "00000000";

memoria (129) <= "00000000";

memoria (130) <= "00000000";

memoria (131) <= "00000000";

memoria (132) <= "00000000";

memoria (133) <= "00000000";

memoria (134) <= "00000000";

memoria (135) <= "00000000";

memoria (136) <= "00000000";

memoria (137) <= "00000000";

memoria (138) <= "00000000";

memoria (139) <= "00000000";

memoria (140) <= "00000000";

memoria (141) <= "00000000";

memoria (142) <= "00000000";

memoria (143) <= "00000000";

memoria (144) <= "00000000";

memoria (145) <= "00000000";

memoria (146) <= "00000000";

memoria (147) <= "00000000";

memoria (148) <= "00000000";

memoria (149) <= "00000000";

memoria (150) <= "00000000";

memoria (151) <= "00000000";

memoria (152) <= "00000000";

memoria (153) <= "00000000";

memoria (154) <= "00000000";

memoria (155) <= "00000000";

memoria (156) <= "00000000";

memoria (157) <= "00000000";

memoria (158) <= "00000000";

memoria (159) <= "00000000";

memoria (160) <= "00000000";

memoria (161) <= "00000000";

memoria (162) <= "00000000";

memoria (163) <= "00000000";

memoria (164) <= "00000000";

memoria (165) <= "00000000";

memoria (166) <= "00000000";

memoria (167) <= "00000000";

memoria (168) <= "00000000";

memoria (169) <= "00000000";

memoria (170) <= "00000000";

memoria (171) <= "00000000";

memoria (172) <= "00000000";

memoria (173) <= "00000000";

memoria (174) <= "00000000";

memoria (175) <= "00000000";

memoria (176) <= "00000000";

memoria (177) <= "00000000";

memoria (178) <= "00000000";

memoria (179) <= "00000000";

memoria (180) <= "00000000";

memoria (181) <= "00000000";

memoria (182) <= "00000000";

memoria (183) <= "00000000";

memoria (184) <= "00000000";

memoria (185) <= "00000000";

memoria (186) <= "00000000";

memoria (187) <= "00000000";

memoria (188) <= "00000000";

memoria (189) <= "00000000";

memoria (190) <= "00000000";

memoria (191) <= "00000000";

memoria (192) <= "00000000";

memoria (193) <= "00000000";

memoria (194) <= "00000000";

memoria (195) <= "00000000";

memoria (196) <= "00000000";

memoria (197) <= "00000000";

memoria (198) <= "00000000";

memoria (199) <= "00000000";

memoria (200) <= "00000000";

memoria (201) <= "00000000";

memoria (202) <= "00000000";

memoria (203) <= "00000000";

memoria (204) <= "00000000";

memoria (205) <= "00000000";

memoria (206) <= "00000000";

memoria (207) <= "00000000";

memoria (208) <= "00000000";

memoria (209) <= "00000000";

memoria (210) <= "00000000";

memoria (211) <= "00000000";

memoria (212) <= "00000000";

memoria (213) <= "00000000";

memoria (214) <= "00000000";

memoria (215) <= "00000000";

memoria (216) <= "00000000";

memoria (217) <= "00000000";

memoria (218) <= "00000000";

memoria (219) <= "00000000";

memoria (220) <= "00000000";

memoria (221) <= "00000000";

memoria (222) <= "00000000";

memoria (223) <= "00000000";

memoria (224) <= "00000000";

memoria (225) <= "00000000";

memoria (226) <= "00000000";

memoria (227) <= "00000000";

memoria (228) <= "00000000";

memoria (229) <= "00000000";

memoria (230) <= "00000000";

memoria (231) <= "00000000";

memoria (232) <= "00000000";

memoria (233) <= "00000000";

memoria (234) <= "00000000";

memoria (235) <= "00000000";

memoria (236) <= "00000000";

memoria (237) <= "00000000";

memoria (238) <= "00000000";

memoria (239) <= "00000000";

memoria (240) <= "00000000";

memoria (241) <= "00000000";

memoria (242) <= "00000000";

memoria (243) <= "00000000";

memoria (244) <= "00000000";

memoria (245) <= "00000000";

memoria (246) <= "00000000";

memoria (247) <= "00000000";

memoria (248) <= "00000000";

memoria (249) <= "00000000";

memoria (250) <= "00000000";

memoria (251) <= "00000000";

memoria (252) <= "00000000";

memoria (253) <= "00000000";

memoria (254) <= "00000000";

memoria (255) <= "00000000";

else

--ESCREVER NA MEMORIA

if(read = '1' and write = '0'))then

out\_mem (15 downto 8) <= memoria (to\_integer(unsigned(end\_mem)));

out\_mem(7 downto 0) <= memoria(to\_integer(unsigned(end\_mem+1)));

--LEITURA DA MEMORIA

elsif (read = '0' and write = '1')) then

memoria(to\_integer(unsigned(end\_mem))) <= in\_mem(15 downto 8);

memoria(to\_integer(unsigned(end\_mem+1))) <= int\_mem(7 downto 0);

end if;

end if;

end process;

end rtl;