

Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.111 - Digital Systems Laboratory
Problem Set #5 SOLUTIONS

Problem 5.1

```
/* kompressor.as */
/* assembly file of the MCU for kompressor-brand car compactors */

# SPEC_FILE = kompressor.sp;
# LIST_FILE = kompressor.lst;
# MASK_COUNT = 8;
# SET_ADDRESS = 0;
# LOAD_ADDRESS = 0;

/* First, reset the Kompressor car compactor to maximum size. */

RELEASE:  IF KOMPRESSOR_IS SIZE_OF_CAR THEN CJMP CONVEYOR;
          ASSERT KOMPRESSOR_RELEASE;
          JMP RELEASE;

/* Next, you add the rest of the code. */

/* First, Clear the Kompressor */
CONVEYOR: IF not_in_kompressor THEN CJMP FILL;
          ASSERT move_conveyor;
          JMP CONVEYOR;

/* Then, put a new car in it */
FILL:     IF in_kompressor THEN CJMP READY;
          ASSERT move_conveyor;
          JMP FILL;

/* now compact until it's as small as possible */
COMPACT:  IF size_of_metal_briefcase THEN CJMP RELEASE;
          ASSERT kompressor_compact;
          JMP COMPACT;
```

```

library ieee;
use ieee.std_logic_1164.all;

entity kompressor_assertion is
  port (op : in std_logic_vector(7 downto 0);
        clk : in std_logic;
        compactor_on : out std_logic;
        compactor_dir : out std_logic;
        -- 1 for compact 0 for release
        conveyor : out std_logic);
end;

architecture arch of kompressor_assertion is
begin
  process(clk)
  begin
    if rising_edge(clk) then
      -- by default, do nothing
      compactor_on <= '0';
      conveyor <= '0';

      if op(7) = '1' then
        -- Assert Instruction

        if op(2 downto 1) = "11" then
          compactor_on <= '1';
          compactor_dir <= '1';
        elsif op (2 downto 1) = "10" then
          compactor_on <= '1';
          compactor_dir <= '0';
        end if;

        if op(0) = '1' then
          conveyor <= '1';
        end if;

      end if;
    end if;
  end process;
end arch;

```

Problem 5.2

The problem is that once the victim detector goes off, it continuously resets the MCU to address 0, preventing the MCU from doing anything to move the car with victim out of the Kompressor. Since the MCU can't do move the car, the victim detector never goes off and the whole thing just sits there.

To fix this, we need to change the victim detector to only reset the MCU once for each victim. We need to detect the rising edge of the victim signal input.

```
library ieee;
use ieee.std_logic_1164.all;

entity kompressor_assertion is
  port (op : in std_logic_vector(7 downto 0);
        clk : in std_logic;
        compactor_on : out std_logic;
        compactor_dir : out std_logic;
        -- 1 for compact 0 for release
        conveyor : out std_logic;

        -- add victim logic
        victim : in std_logic;
        clear_mcu : out std_logic;

  );
end;

architecture arch of kompressor_assertion is
begin

  signal victim_old : std_logic := '0';      -- for catching edge

  process(clk)
  begin
    if rising_edge(clk) then
      -- by default, do nothing
      compactor_on <= '0';
      conveyor <= '0';

      if op(7) = '1' then
        -- Assert Instruction

        if op(2 downto 1) = "11" then
          compactor_on <= '1';
          compactor_dir <= '1';
        elsif op (2 downto 1) = "10" then
```

```
        compactor_on <= '1';
        compactor_dir <= '0';
    end if;

    if op(0) = '1' then
        conveyor <= '1';
    end if;

end if;

-- victim logic
victim_old <= victim;
clear_mcu <= '0';
if (victim_old = '0' and victim = '1') then
    clear_mcu <= '1'
end if;

end if;
end process;
end arch;
```