한국항해항만학회 제30권 제2호 추계학술대회논문집 pp. 69~73, 2006. 12

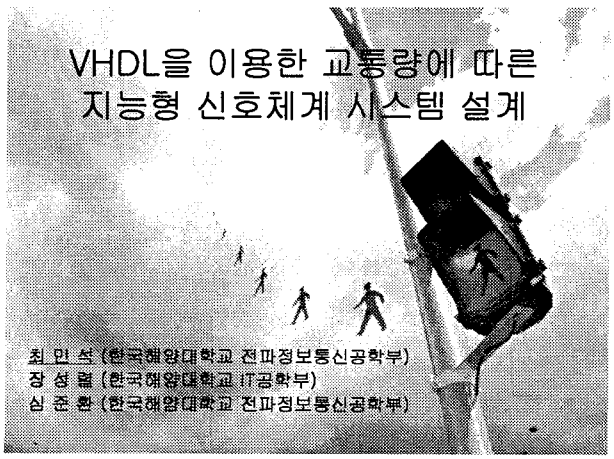
VHDL을 이용한 교통량에 따른 지능형 신호체계 시스템 설계

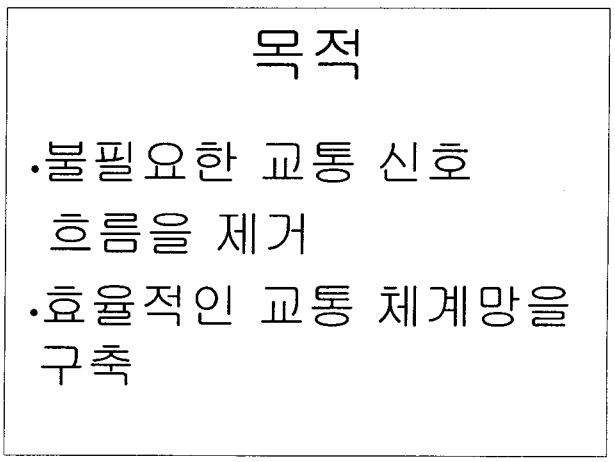
최 민석* · 장 성렬** · 심 준환*

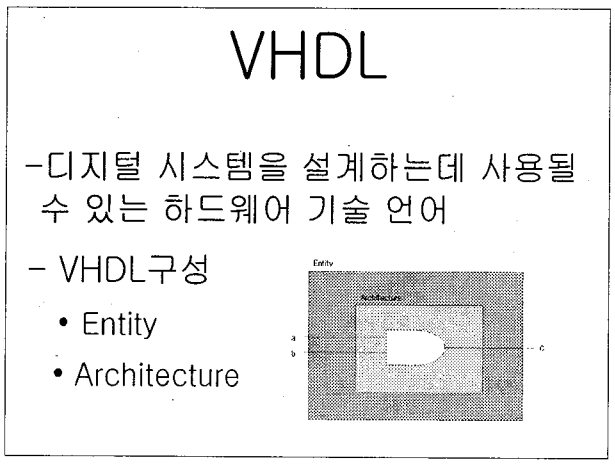
*한국해양대학교 전파 · 정보통신공학부, **한국해양대학교 IT공학부

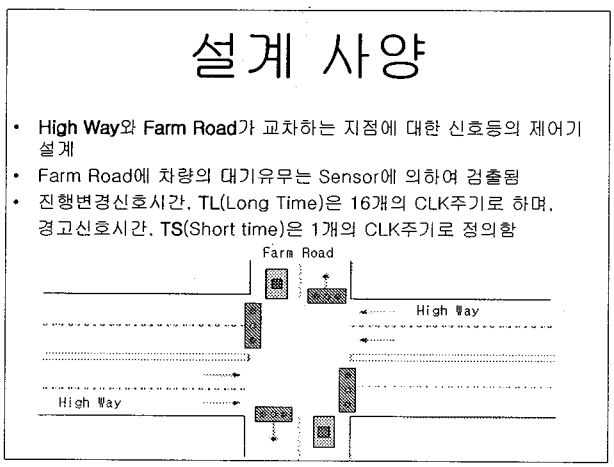
요 약 : 일반신호기는 정해진 신호시간에 따라 독립적으로 운영되는 정주기신호기와 TOD(Time of Day)식 신호기이다. 이런 일반신호기의 단점으로는 변동되는 교통량에 대응이 불가능하고, 교통량에 관계없이 연속적으로 신호체계가 이루어지기 때문에 차량의 이동성이 비효율적으로 됨에 따라 생산성 감소, 에너지낭비 및 자동차 배기가스 증가를 초래하고 있다. 이를 개선하기 위한 방법으로서 본 논문에서는 4차선의 주차로와 2차선의 소방로로 이루어진 교차로에서 주차로의 교통량이 많을 경우, 차로에서는 녹색불이 지속되어 교통의 흐름을 원활하게 해주며, 소방로에서 차량이 일정량 증가하게 되면, 주차로의 신호등이 적색등으로 바꾸어 소방로의 차량이 교통되도록 하는 지능형 교통체계에 대해 VHDL을 이용하여 지능형 신호체계 시스템을 설계하였다.

핵심용어 : 신호등, VHDL, 지능형 교통체계, FPGA, 적외선센서



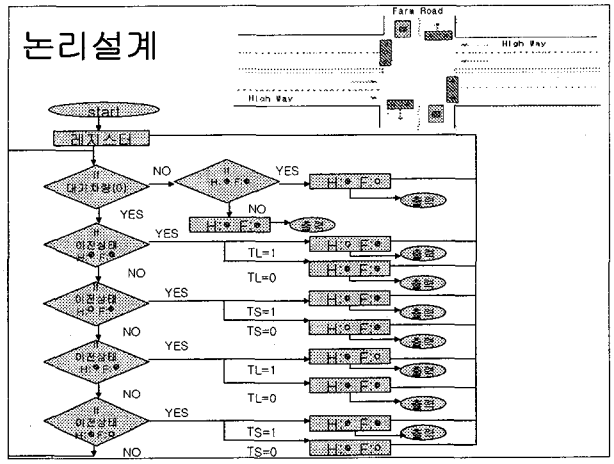




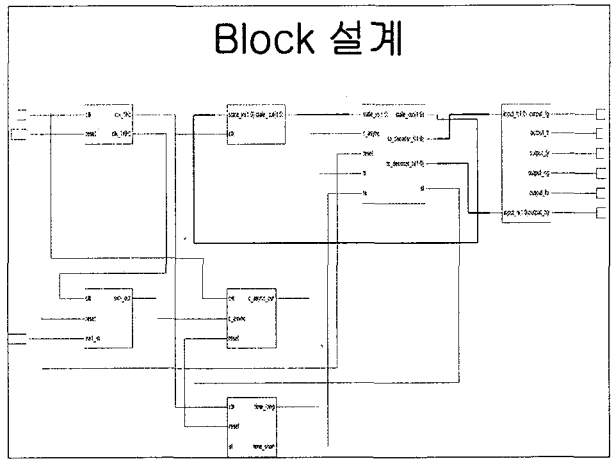


* 대표저자 : 최민석(학생회원) spooky007@hanmail.net
* 교신저자 : 심준환(정회원) jhsim@hhu.ac.kr

논리설계

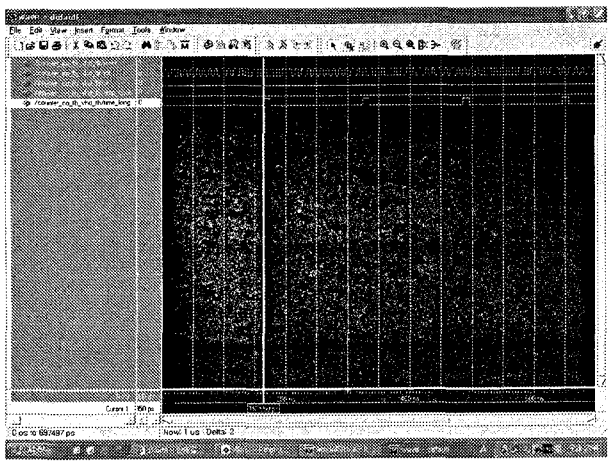


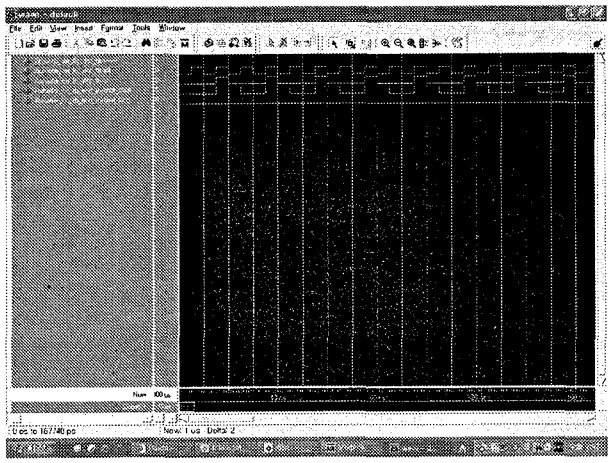
Block 설계



```
--- 파일명 : counter.vhd
--- 설명   : TL 시간과 TS 시간을 결정하는 카운터

library ieee;
use ieee.std_logic_1164.all;

entity counter is
        port(
                reset : in   std_logic;
                clk   : in   std_logic;
                st    : in   std_logic;
                time_short : out      std_logic;
                time_long  : out      std_logic
        );
end counter;

architecture main of counter is
signal count : std_logic_vector(3 downto 0);
begin

        process (clk, reset, st)
        begin

                -- reset(초기화) 상태
                if reset = '0' then
                        time_short <= '0';
                        time_long  <= '0';
                        count <= "0000";

                -- 일반적인 상태
                else
                if (clk'event) and (clk='1') then
                -- Yellow 상태인 경우
                if st = '1' then
                count <= "0000";
                time_long  <= '0';
                time_short <= '1';
```

```
-- Green 혹은 Red 상태인 경우
else
-- TL 시간이 경과
if count = "1111" then
time_long  <= '1';
time_short <= '0';
count      <= "0000";
else
time_long  <= '0';
time_short <= '0';

case count is
  when "0000" => count <= "0001";
  when "0001" => count <= "0010";
  when "0010" => count <= "0011";
  when "0011" => count <= "0100";
  when "0100" => count <= "0101";
  when "0101" => count <= "0110";
  when "0110" => count <= "0111";
  when "0111" => count <= "1000";
  when "1000" => count <= "1001";
  when "1001" => count <= "1010";
  when "1010" => count <= "1011";
  when "1011" => count <= "1100";
  when "1100" => count <= "1101";
  when "1101" => count <= "1110";
  when "1110" => count <= "1111";
  when "1111" => count <= "0000";
  when others => count <= "0000";
    end case;
    end if;
    end if;
    end if;
    end if;

end process;
end main;
```

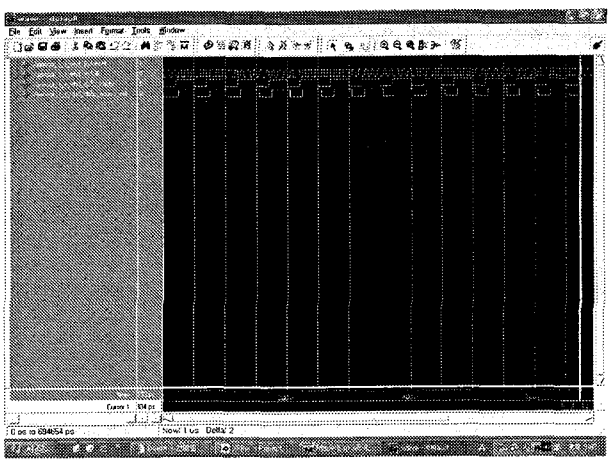


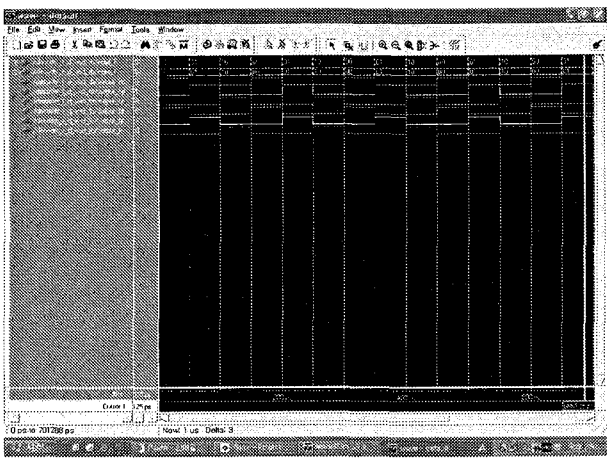


```
-- 파일명 : decoder.vhd
-- 설명   : 들어오는 Highway Light, FarmRoad Light 를
--          Green, Yellow, Red 신호로 각각 나뉘어서 내보내는 모듈

library ieee;
use ieee.std_logic_1164.all;

-- 001 : Green  Light
-- 010 : Yellow Light
-- 100 : Red    Light

entity decoder is
        port(
                input_h     : in        std_logic_vector(1 downto 0);
                input_f     : in        std_logic_vector(1 downto 0);
                output_hg   : out       std_logic;
                output_hy   : out       std_logic;
                output_hr   : out       std_logic;
                output_fg   : out       std_logic;
                output_fy   : out       std_logic;
                output_fr   : out       std_logic
        );
end decoder;

architecture main of decoder is
begin

        process (input_h, input_f)
        begin

                -- Highway Green
                if input_h = "00" then
                        output_hg <= '1';
                        output_hy <= '0';
                        output_hr <= '0';

                -- Highway Yellow
                elsif input_h = "01" then
                        output_hg <= '0';
                        output_hy <= '1';
                        output_hr <= '0';
```

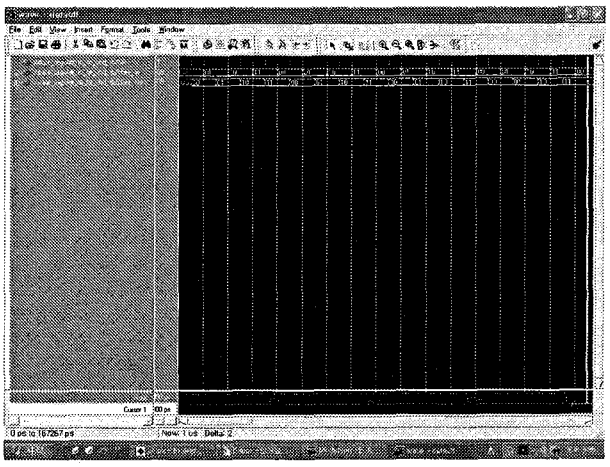
```
-- 파일명 : sensor.vhd
-- 설명   : 차량이 들어오는 지 검사하는 모듈

library ieee;
use ieee.std_logic_1164.all;

entity sensor is
        port(
                reset       : in        std_logic;
                clk         : in        std_logic;
                c_async     : in        std_logic;
                c_async_out             : out       std_logic
        );
end sensor;

architecture main of sensor is
begin

        process(clk, reset, c_async)
        begin
                if reset = '0' then
                        c_async_out <= '0';
                elsif reset = '1' then
                        if (clk'event) and (clk='1') then
                                -- 차량이 대기중이 아닐 때
                                if c_async = '0' then
                                        c_async_out <= '0';

                                -- 차량이 대기중일 때
                                elsif c_async = '1' then
                                        c_async_out <= '1';
                                end if;
                        end if;
                end if;
        end process;

end main;
```

```
                        -- Highway Red
                elsif input_h = "10" then
                        output_hg <= '0';
                        output_hy <= '0';
                        output_hr <= '1';

                else
                        output_hg <= '0';
                        output_hy <= '0';
                        output_hr <= '0';
                end if;

                -- FarmRoad Green
                if input_f = "00" then
                        output_fg <= '1';
                        output_fy <= '0';
                        output_fr <= '0';

                -- FarmRoad Yellow
                elsif input_f = "01" then
                        output_fg <= '0';
                        output_fy <= '1';
                        output_fr <= '0';

                -- FarmRoad Red
                elsif input_f = "10" then
                        output_fg <= '0';
                        output_fy <= '0';
                        output_fr <= '1';

                else
                        output_fg <= '0';
                        output_fy <= '0';
                        output_fr <= '0';

                end if;

        end process;

end main;
```





```
-- 파일명 : state_register.vhd
-- 설명   : 신호등의 현재 상태를 임시저장하는 레지스터

library ieee;
use ieee.std_logic_1164.all;

-- 00 : Highway Green , FarmRoad Red
-- 01 : Highway Yellow, FarmRoad Red
-- 10 : Highway Red   , FarmRoad Green
-- 11 : Highway Red   , FarmRoad Yellow

entity state_register is
        port(
                clk         : in        std_logic;
                state_in    : in        std_logic_vector(1 downto 0);
                state_out   : out       std_logic_vector(1 downto 0)
        );
end state_register;

architecture main of state_register is
begin

        process (clk, state_in)
        begin
                if (clk'event) and (clk='1') then
                        state_out <= state_in;
                end if;
        end process;

end main;
```



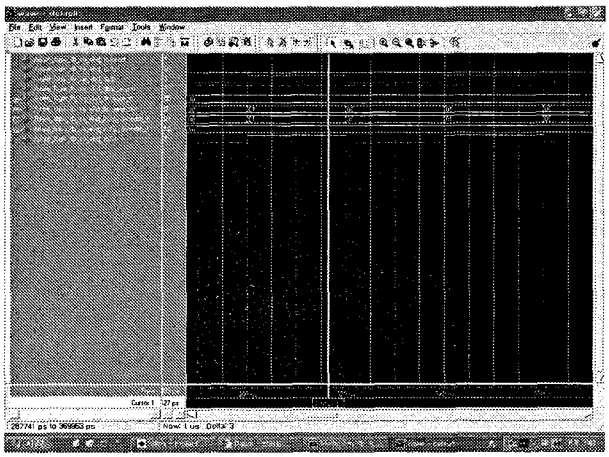
```
                                        -- HR, FG
                                elsif state_in = "10" then
                                        -- TL 시간이 경과하면
                                        if tl = '1' then
                                                to_decoder_h <= "10";
                                                to_decoder_f <= "01";
                                                state_out <= "11";
                                                st <= '1';
                                        else
                                                -- TL 시간 도달전이고 차량이 있으면
                                                if c_async = '1' then
                                                        to_decoder_h <= "10";
                                                        to_decoder_f <= "00";
                                                        state_out <= "10";
                                                        st <= '0';
                                                -- TL 시간 도달전이지만 차량이 없으면
                                                else
                                                        to_decoder_h <= "10";
                                                        to_decoder_f <= "01";
                                                        state_out <= "11";
                                                        st <= '1';
                                                end if;
                                        end if;
                                --HR, FY
                                elsif state_in = "11" then
                                        -- TS 시간이 경과하면
                                        if ts = '1' then
                                                to_decoder_h <= "00";
                                                to_decoder_f <= "10";
                                                state_out <= "00";
                                                st <= '0';
                                        -- TS 시간이 경과하지 않으면
                                        else
                                                to_decoder_h <= "10";
                                                to_decoder_f <= "01";
                                                state_out <= "11";
                                                st <= '0';
                                        end if;
                                end if;
                        end if;
                end process;
        end main;
```

```
-- 파일명 : state_logic.vhd
-- 설명 : 레지스터의 현재상태를 가져오고 TL, TS에 따라서 다음 상태를 결정하는 논리 모듈
farmway Red          00 : Green
farmway Red          01 : Yellow
farmway Green        10: Red
farmway Yellow
library ieee;
use ieee.std_logic_1164.all;

entity state_logic is
        port(
                reset                   : in    std_logic;
                ts                      : in    std_logic;
                tl                      : in    std_logic;
                c_async                 : in    std_logic;
                state_in    : in    std_logic_vector(1 downto 0);
                state_out   : out   std_logic_vector(1 downto 0);
                to_decoder_h            : out   std_logic_vector(1 downto 0);
                to_decoder_f            : out   std_logic_vector(1 downto 0);
                st                      : out   std_logic
        );
end state_logic;

architecture main of state_logic is
signal state_tmp : std_logic_vector(1 downto 0);
begin

        process (reset, ts, tl, c_async, state_in)
        begin
-- 초기 셋팅값
if reset = '0' then
to_decoder_h <= "00";
to_decoder_f <= "10";
state_out <= "00";
st <= '0';
end if;
```

--00 : High way Red , farmway Green
--01 : High way Yellow , farmway Red
--10 : High way Green , farmway Red
--11 : High way Red , farmway Yellow

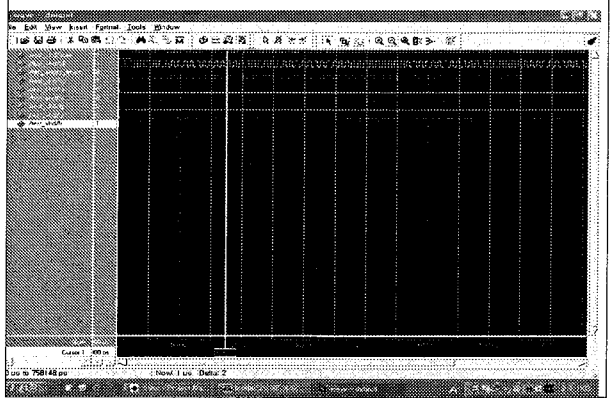
00 : Green
01 : Yellow
10: Red

```
                if reset = '1' then

                -- HG, FR
                if state_in = "00" then
                        -- TL 시간이 경과하고 차량이 대기중
                        if tl = '1' and c_async = '1' then
                                to_decoder_h <= "01";
                                to_decoder_f <= "10";
                                state_out <= "01";
                                st <= '1';
                        -- TL시간이 경과하지 않거나, TL 시간이 경과하고 차량은 없는 경우
                        else
                                to_decoder_h <= "00";
                                to_decoder_f <= "10";
                                state_out <= "00";
                                st <= '0';
                        end if;

                -- HY, FR
                elsif state_in = "01" then
                        -- TS 시간이 경과하면
                        if ts = '1' then
                                to_decoder_h <= "10";
                                to_decoder_f <= "00";
                                state_out <= "10";
                                st <= '0';
                        -- TS 시간이 경과하지 않으면
                        else
                                to_decoder_h <= "01";
                                to_decoder_f <= "10";
                                state_out <= "01";
                                st <= '0';
                        end if;
```

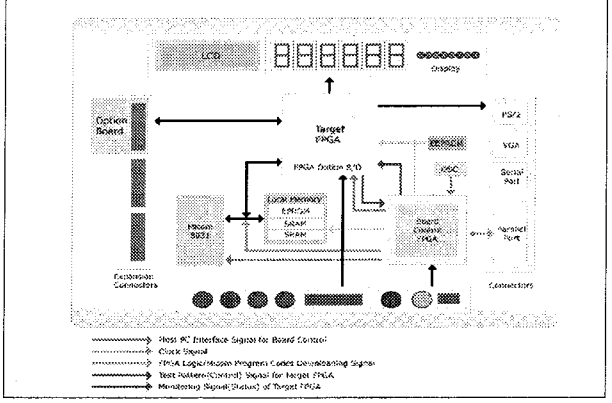


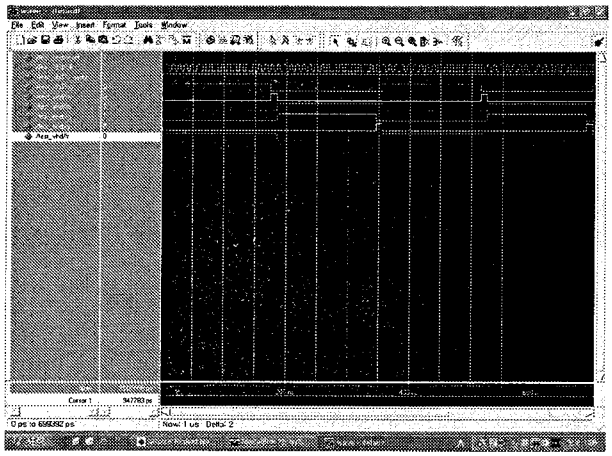
```
-- 파일명 : fsm.vhd
-- 설명 : 프로젝트 메인 파일이며 모든 컴포넌트를 포함한다.

library ieee;
use ieee.std_logic_1164.all;

entity fsm is
        port(
                reset       : in std_logic;
                clk         : in std_logic;
                c_async             : in std_logic;
                hg          : out  std_logic;
                hy          : out  std_logic;
                hr          : out  std_logic;
                fg          : out  std_logic;
                fy          : out  std_logic;
                fr          : out  std_logic
        );
end fsm;

architecture main of fsm is
        component state_register
                port(
                        clk     : in  std_logic;
                        state_in  : in        std_logic_vector(1 downto 0);
                        state_out   : out     std_logic_vector(1 downto 0)
                );
        end component;
        component counter is
                port(
                        reset   : in std_logic;
                        clk : in std_logic;
                        st : in  std_logic;
                        time_short  : out std_logic;
                        time_long   : out std_logic
                );
        end component;
        component sensor is
                port(
                        reset       : in std_logic;
                        clk         : in  std_logic;
                        c_async     : in  std_logic;
                        c_async_out             : out  std_logic
                );
```

최종 시뮬레이션

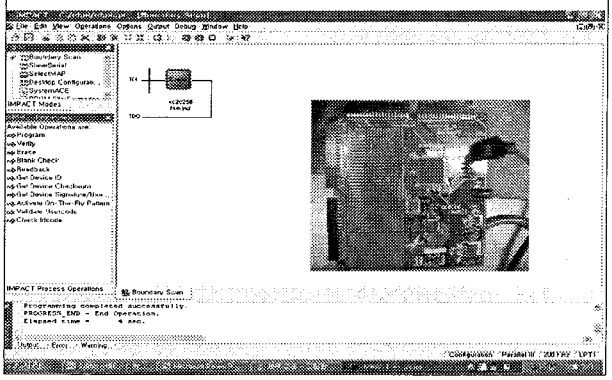


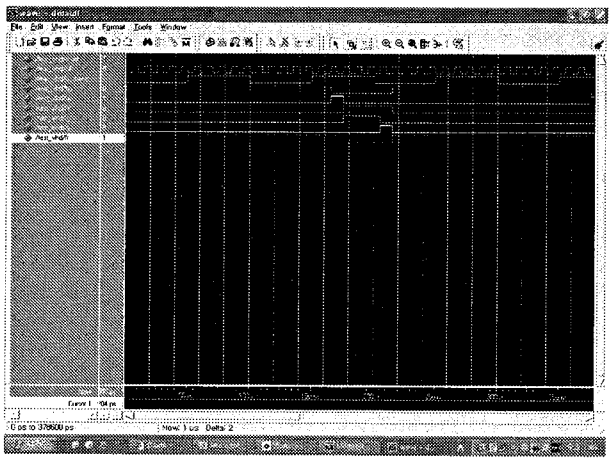
신호체계 시스템 구성도





CPLD 다운로드





센서