FPGA LPM Component
Stop-Watch Counter

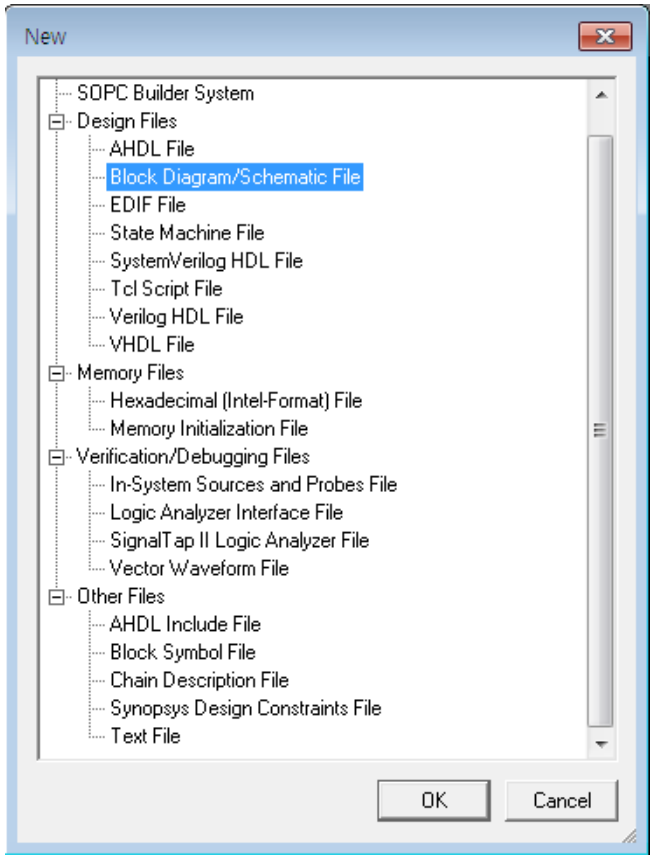
JAICHANGPARK

20181017

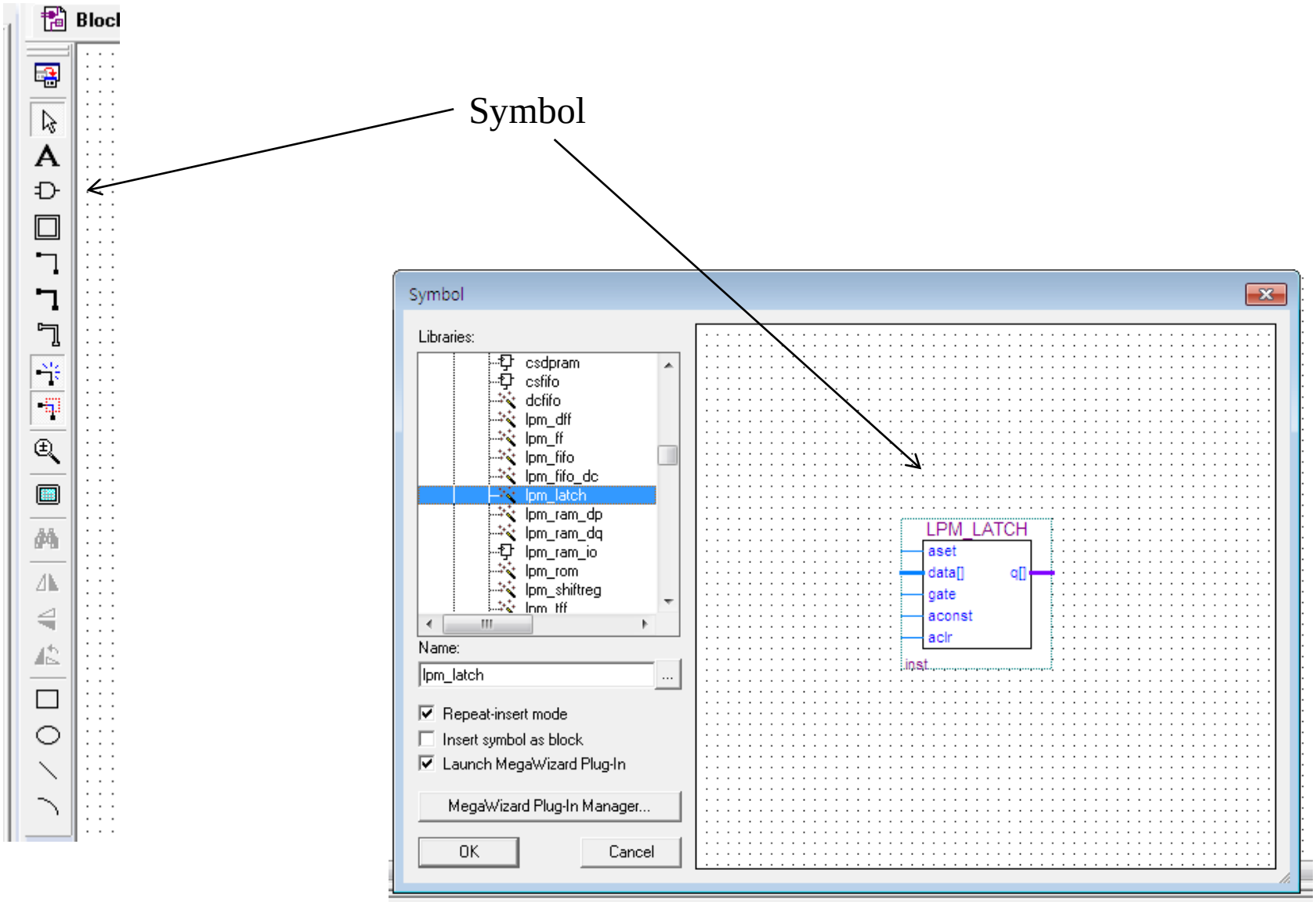
Aim

- Use the FPGA to create a stopwatch counter.
- Use the internal clock of the DE2 board to create a frequency divider counter and check the result.
- Create a counter that increments every second..

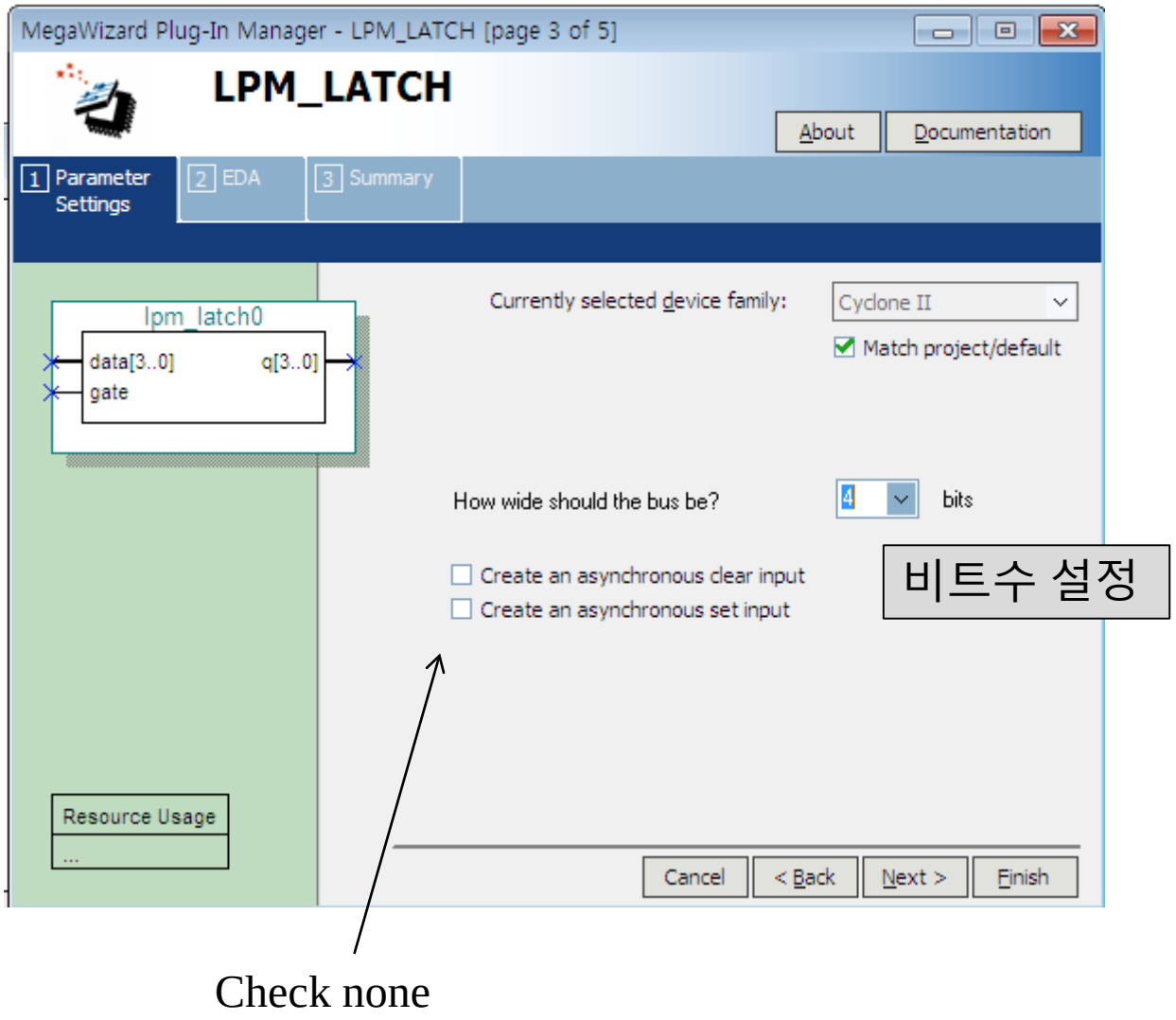
- Create new File



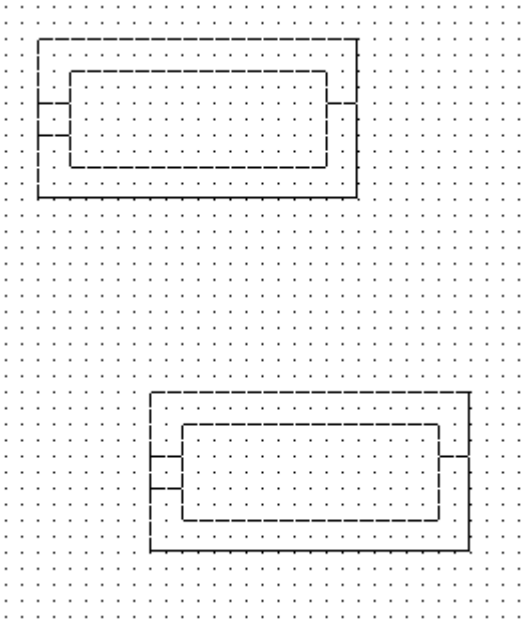
- Choose new Symbol block

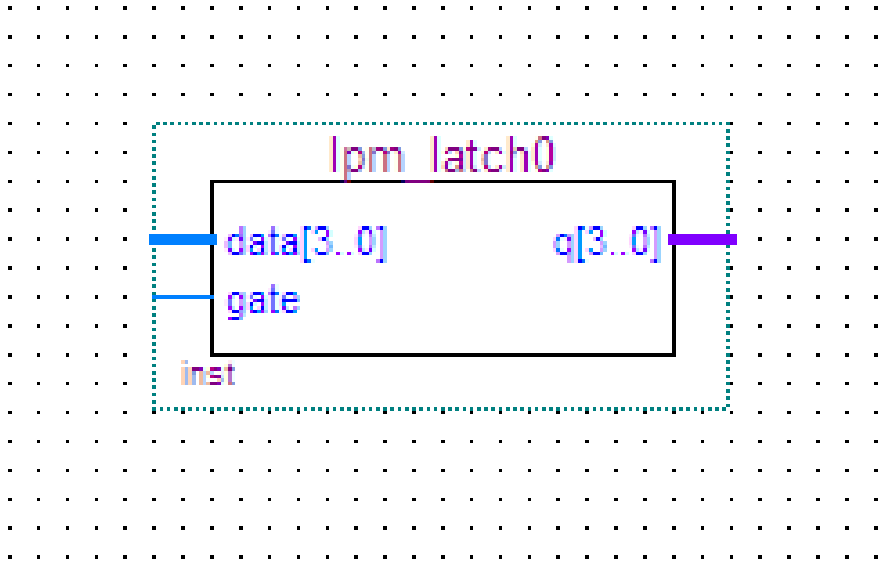


➢ Mega Wizard plug- in manager



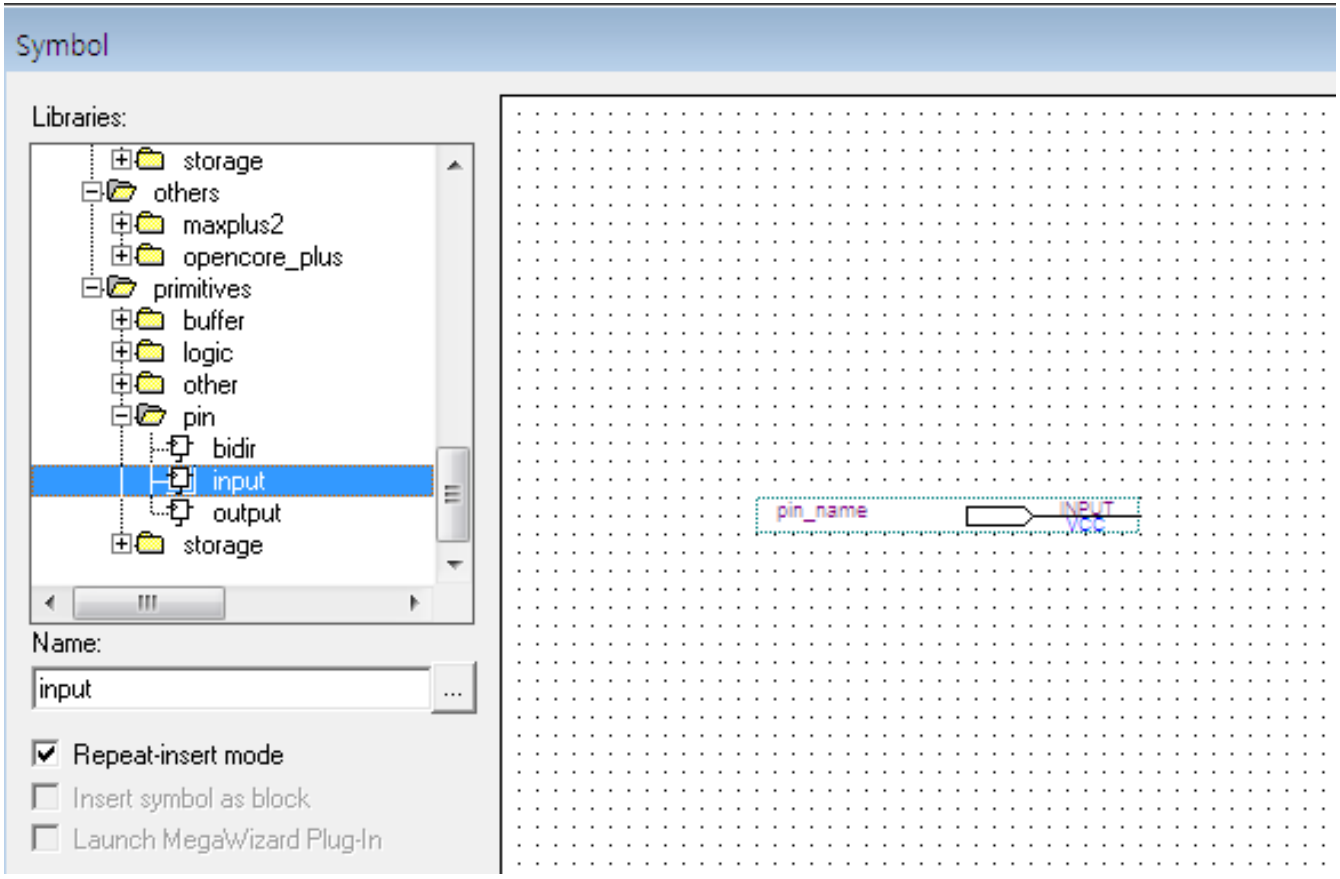
➢ 배치 영역 설정



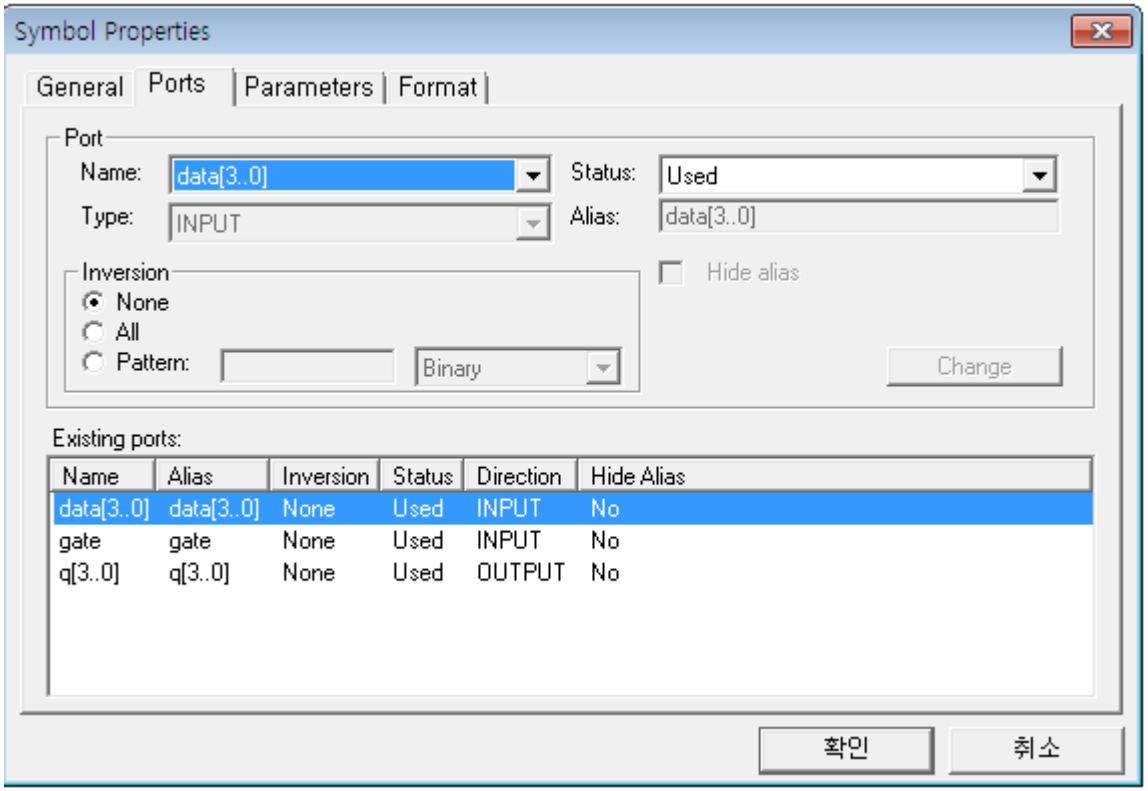


Place the set lpm block on the sketch.
First of all, I created a try block by practice and put it on my sketch.

- Create input / output pin using lpm symbol manager

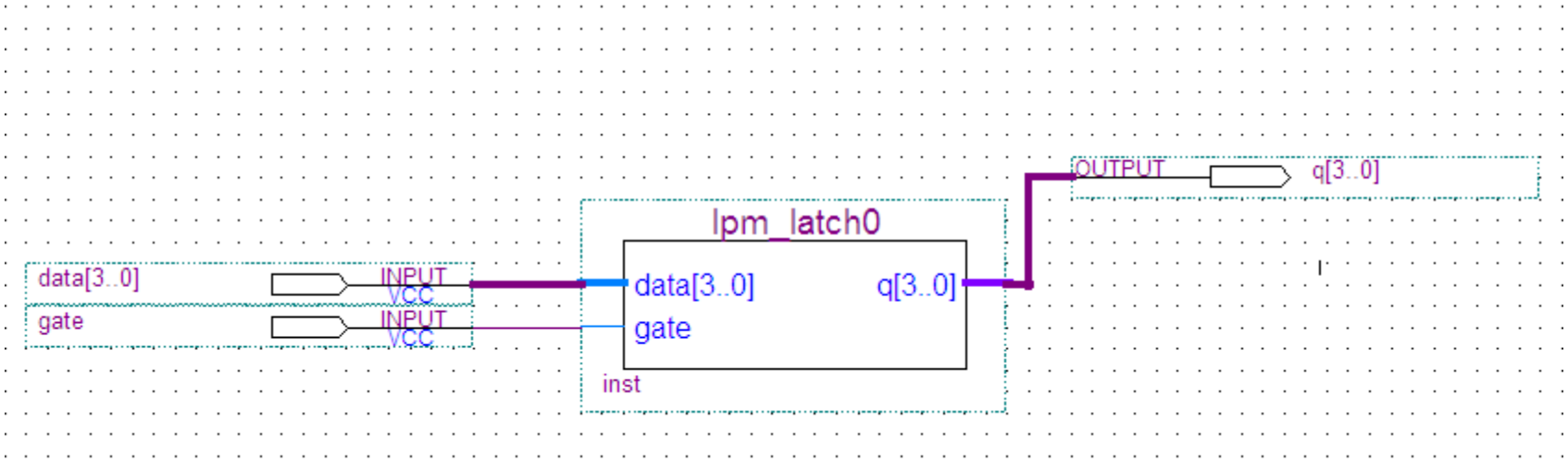


- Check and set symbol port properties

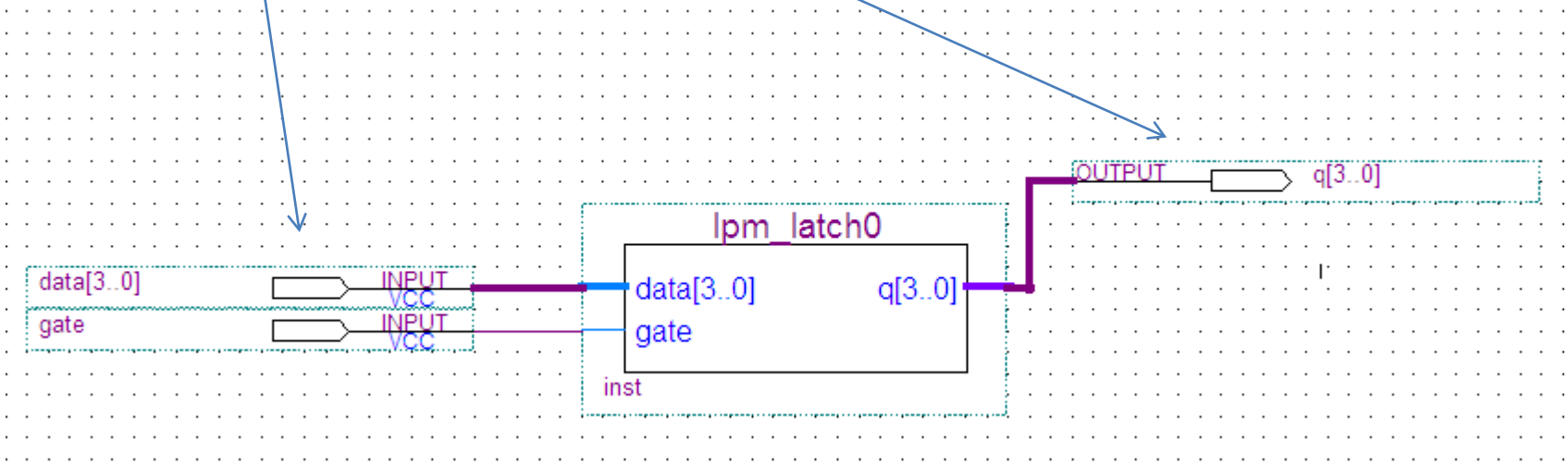


After creating the symbol, you can see the set information by entering the attribute of the block.

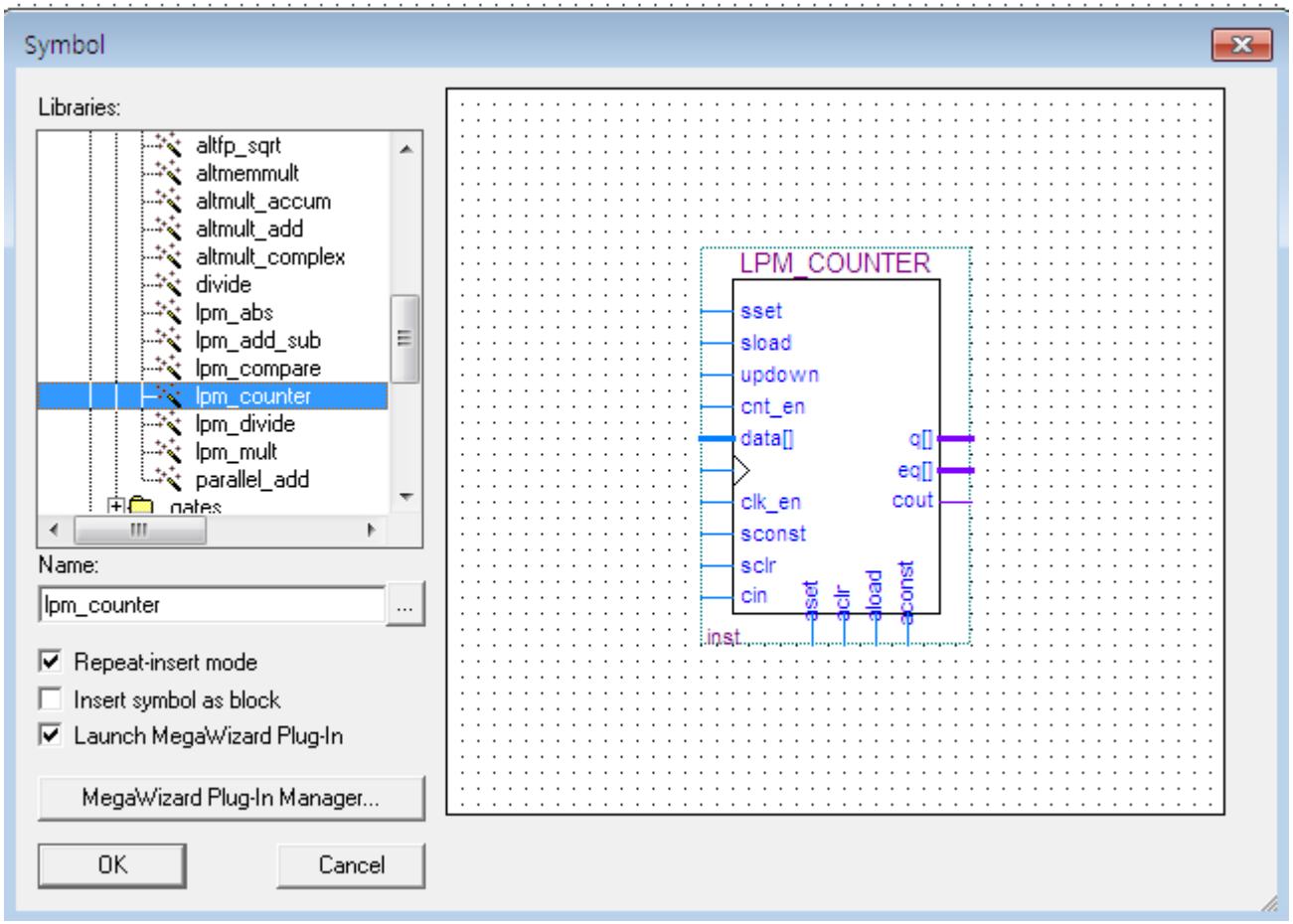
- Connection between block and input



```
ENTITY lpm_latch0 IS
    PORT
    (
        data        : IN STD_LOGIC_VECTOR (3 DOWNTO 0);
        gate        : IN STD_LOGIC ;
        q           : OUT STD_LOGIC_VECTOR (3 DOWNTO 0)
    );
END lpm_latch0;
```

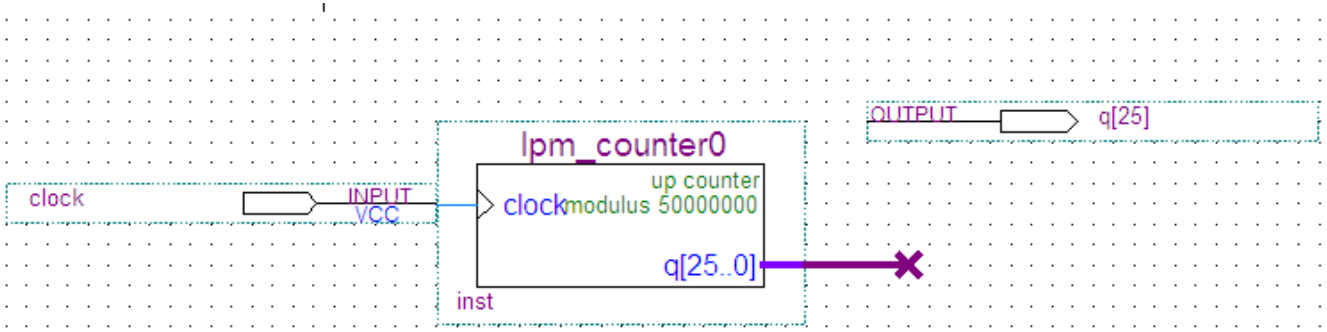


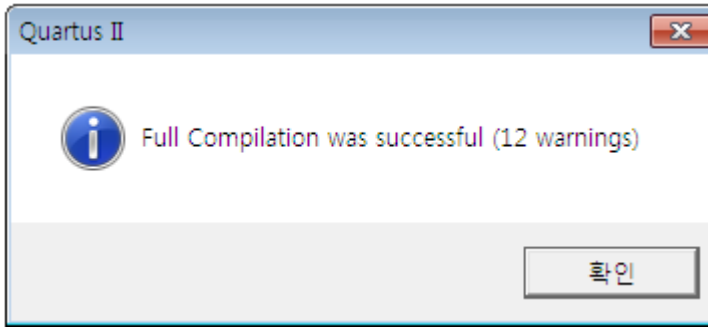
- Lpm_counter


Symbol
Libraries:
altfp_sqrt
altmemmult
altmult_accum
altmult_add
altmult_complex
divide
lpm_abs
lpm_add_sub
lpm_compare
lpm_counter
lpm_divide
lpm_mult
parallel_add
gates
Name:
lpm_counter
Repeat-insert mode
Insert symbol as block
Launch MegaWizard Plug-In
MegaWizard Plug-In Manager...
OK
Cancel
LPM_COUNTER
sset
sload
updown
cnt_en
data[]
clk_en
sconst
sclr
cin
q[]
eq[]
cout
aset
aclr
aload
aconst
inst

➢ Create a clock divide counter

A counter that divides the internal frequency of 50MHz is needed.



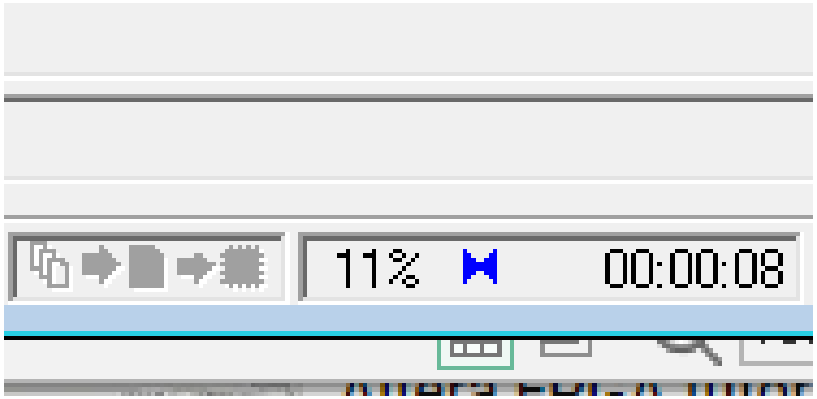


The division ratio for switching from 50 MHz to 1 Hz is 50 million.
So to get a 1Hz waveform with a duty ratio of 50%, set the count value to 50,000 / 2 - 1 = 24,999,999. This count value is 0x17D 783F in hexadecimal, and is 25 bits.

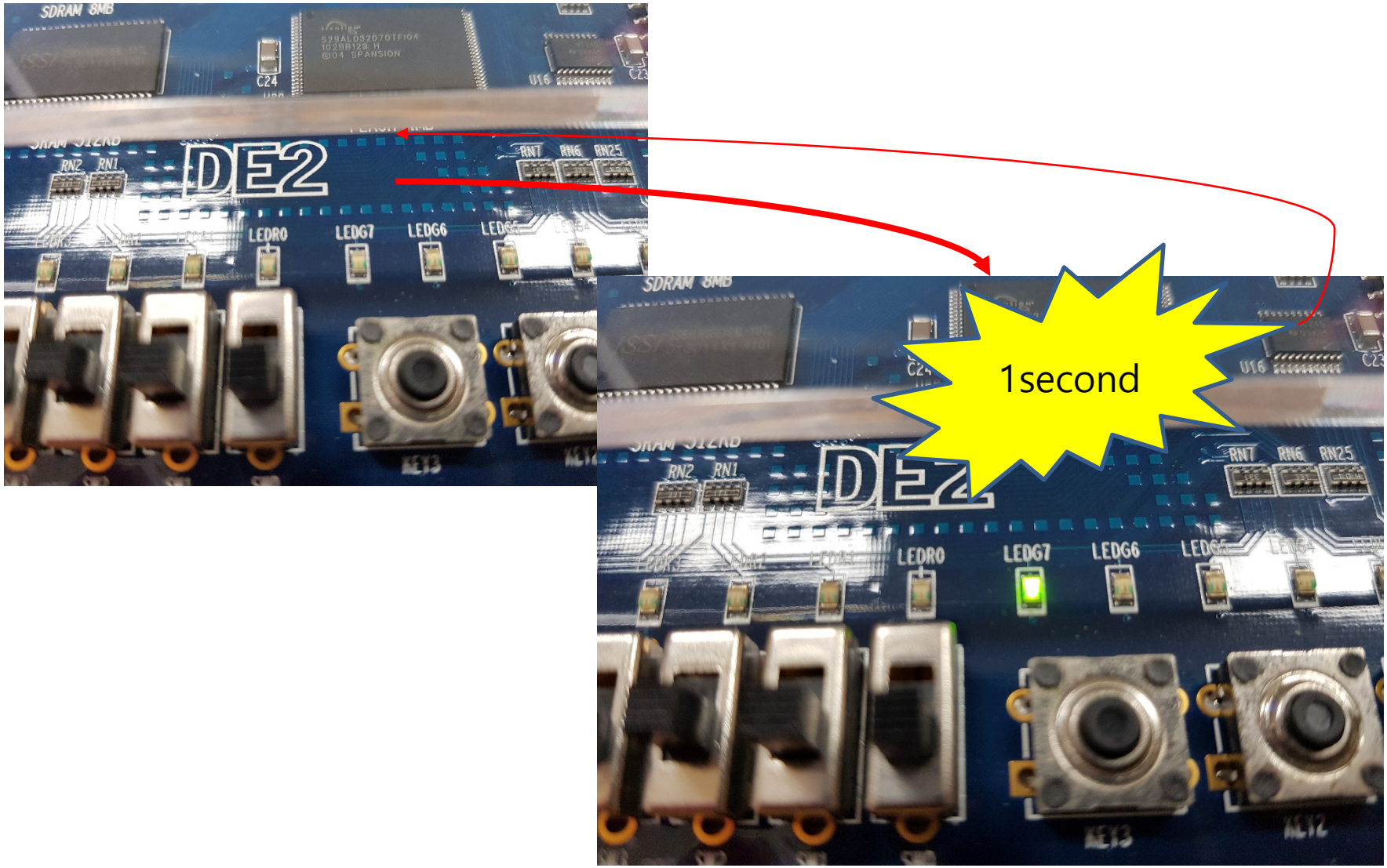
➢ 핀 포팅후 재 컴파일

Edit: PIN_N2

	From	To	Assignment Name	Value	Enabled
1		|	Partition Hierarchy	root_partition	Yes
2		clock	Location	PIN_N2	Yes
3		q[25]	Location	PIN_AE22	Yes
4		q	Location		Yes
5	<<new>>	<<new>>	<<new>>		

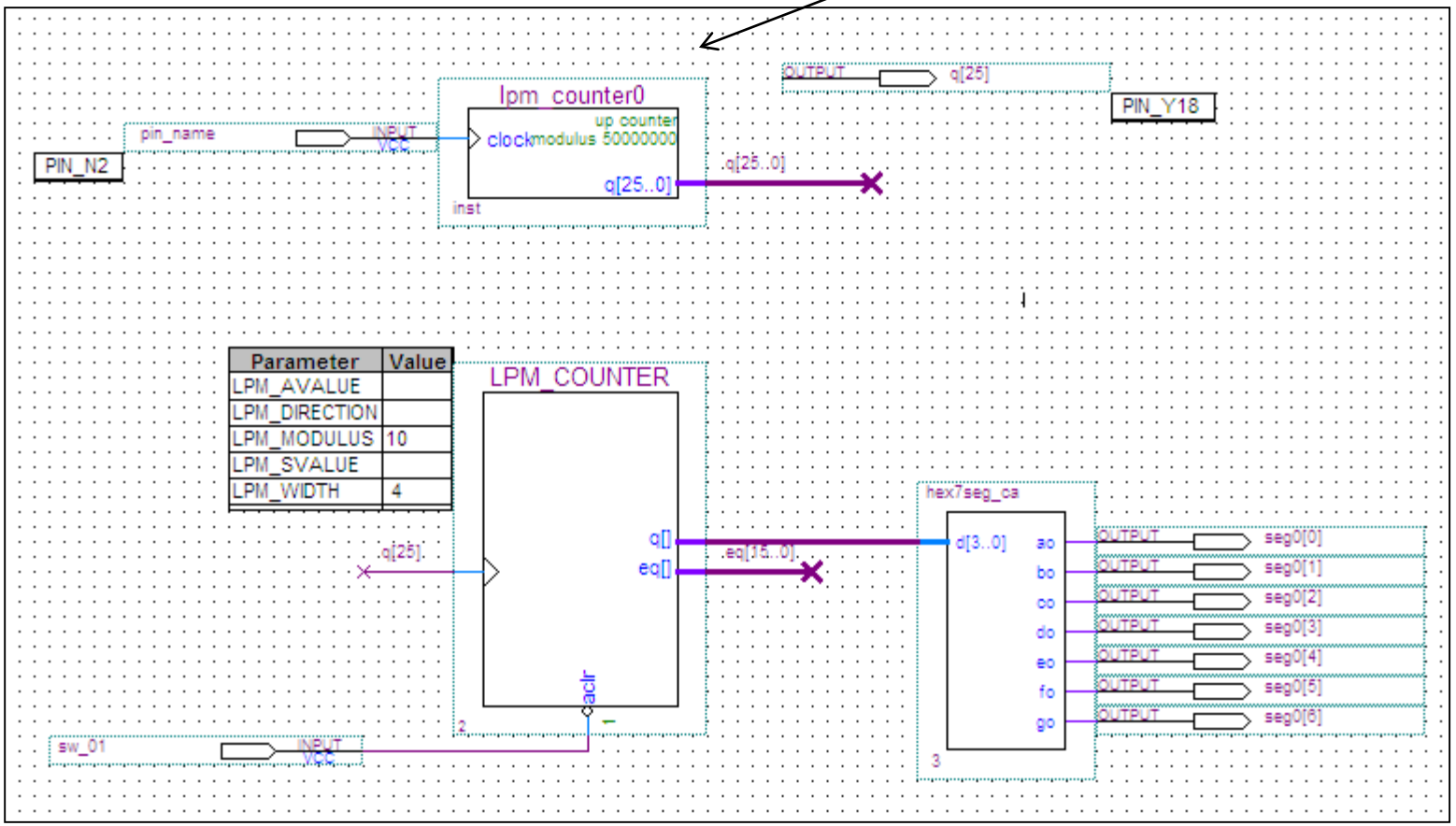


- Result of a clock divide counter

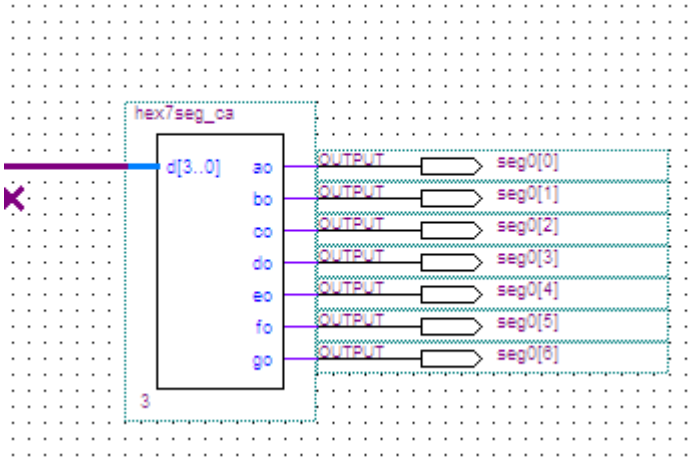


- **Stopwatch primary block schematic**

클럭 분주는 26비트 카운터사용

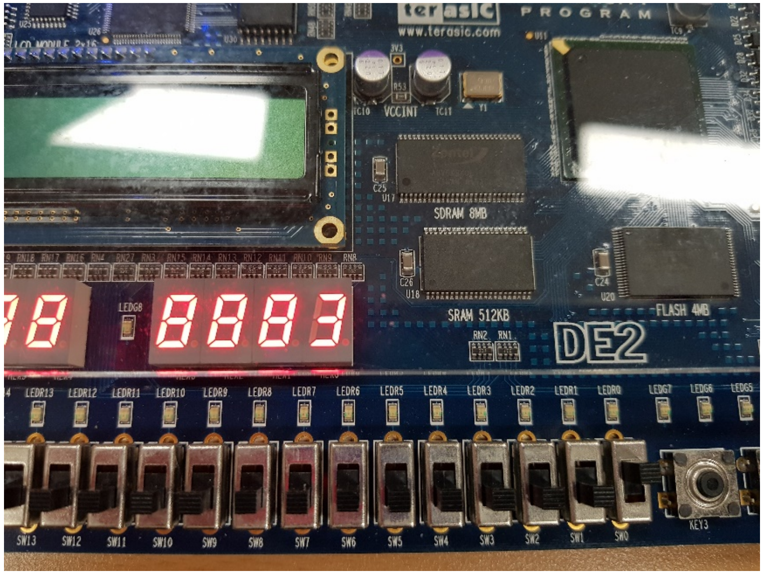


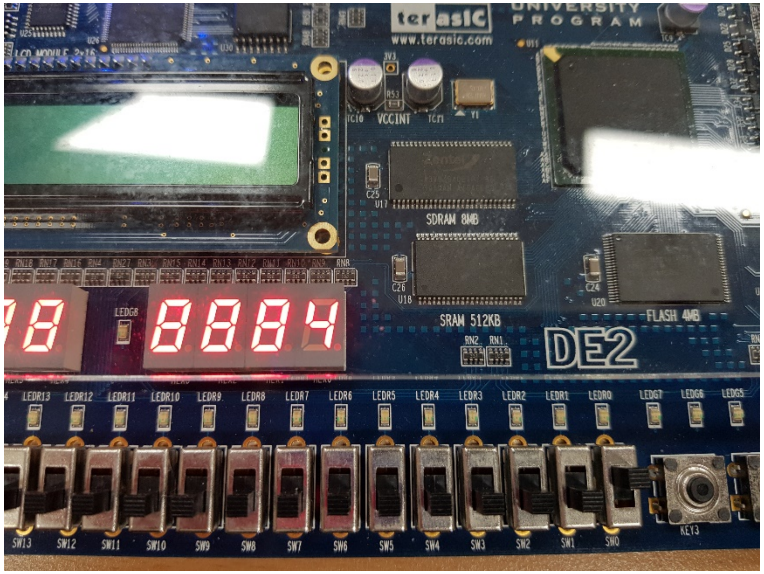
- Segment controller block production for segment manipulation

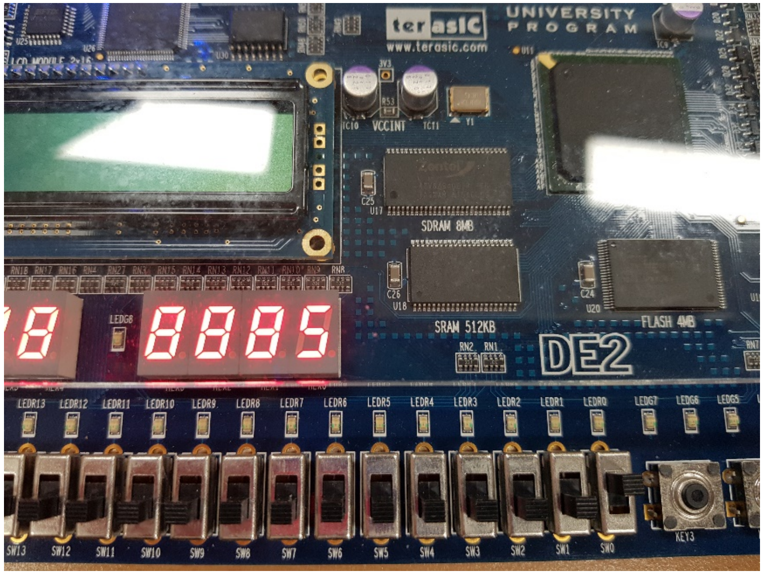


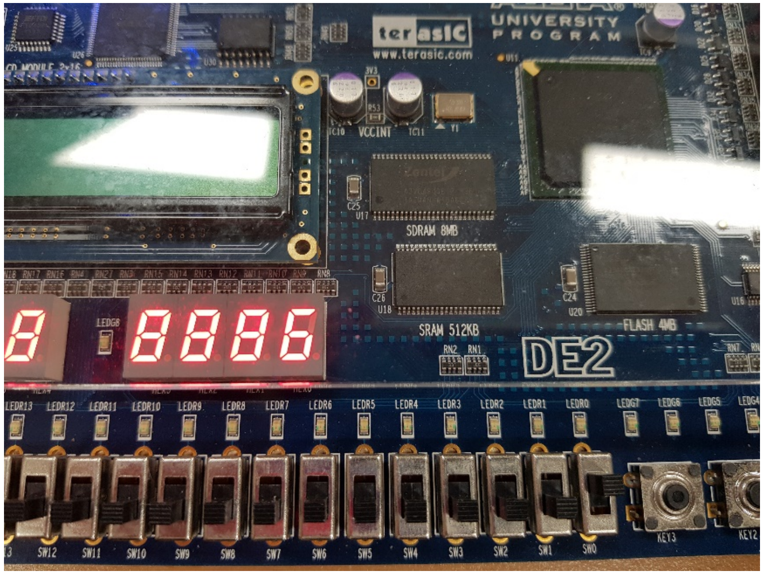

```
ARCHITECTURE seven_segment OF hex7seg_ca IS
    SIGNAL input : STD_LOGIC_VECTOR (3 downto 0);
    SIGNAL output: STD_LOGIC_VECTOR (6 downto 0);
BEGIN
    WITH d SELECT
        output <=   "0000001" WHEN "0000", -- display 0
                    "1001111" WHEN "0001", -- display 1
                    "0010010" WHEN "0010", -- display 2
                    "0000110" WHEN "0011", -- display 3
                    "1001100" WHEN "0100", -- display 4
                    "0100100" WHEN "0101", -- display 5
                    "0100000" WHEN "0110", -- display 6 (with tail)
                    "0001111" WHEN "0111", -- display 7
                    "0000000" WHEN "1000", -- display 8
                    "0000100" WHEN "1001", -- display 9 (with tail)
                    "0001000" WHEN "1010", -- display A
                    "1100000" WHEN "1011", -- display b
                    "0110001" WHEN "1100", -- display C
                    "1000010" WHEN "1101", -- display d
                    "0110000" WHEN "1110", -- display E
                    "0111000" WHEN "1111", -- display F
                    "1111111" WHEN others;
```

➢ **Stopwatch primary block Result**

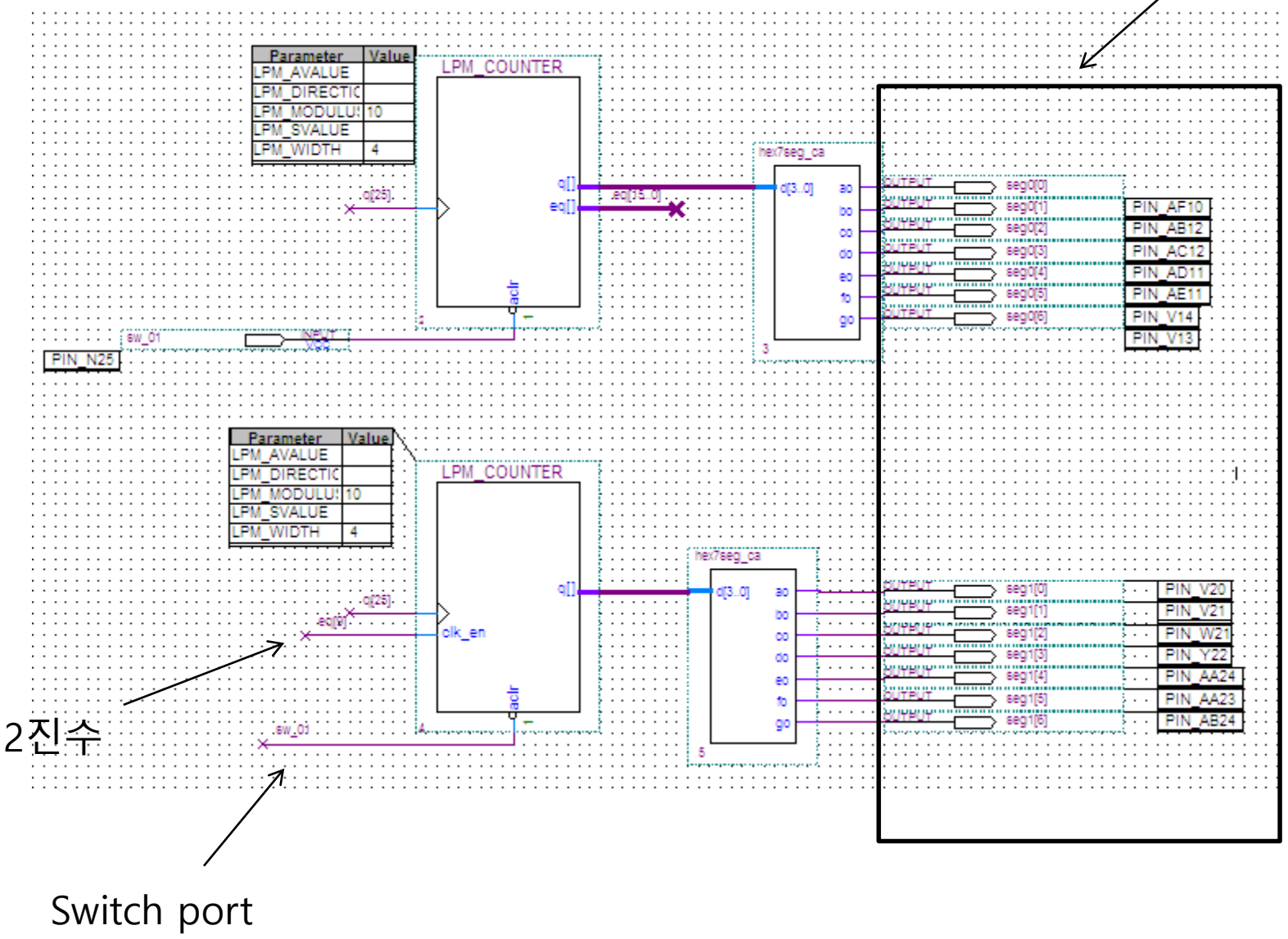








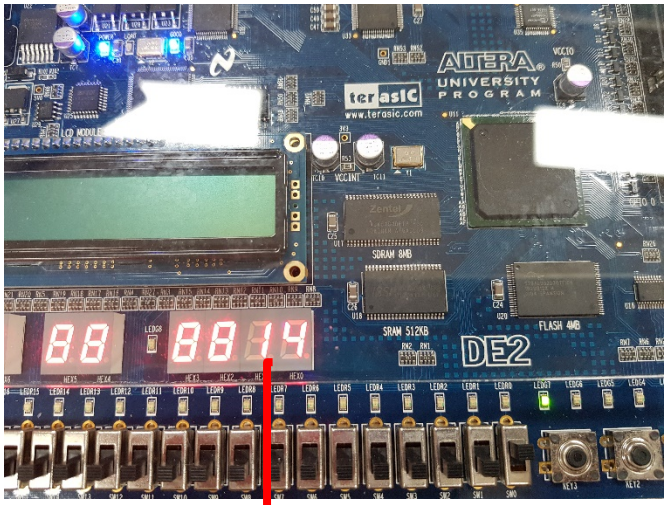
- **Stopwatch Secondary Block Schematic**

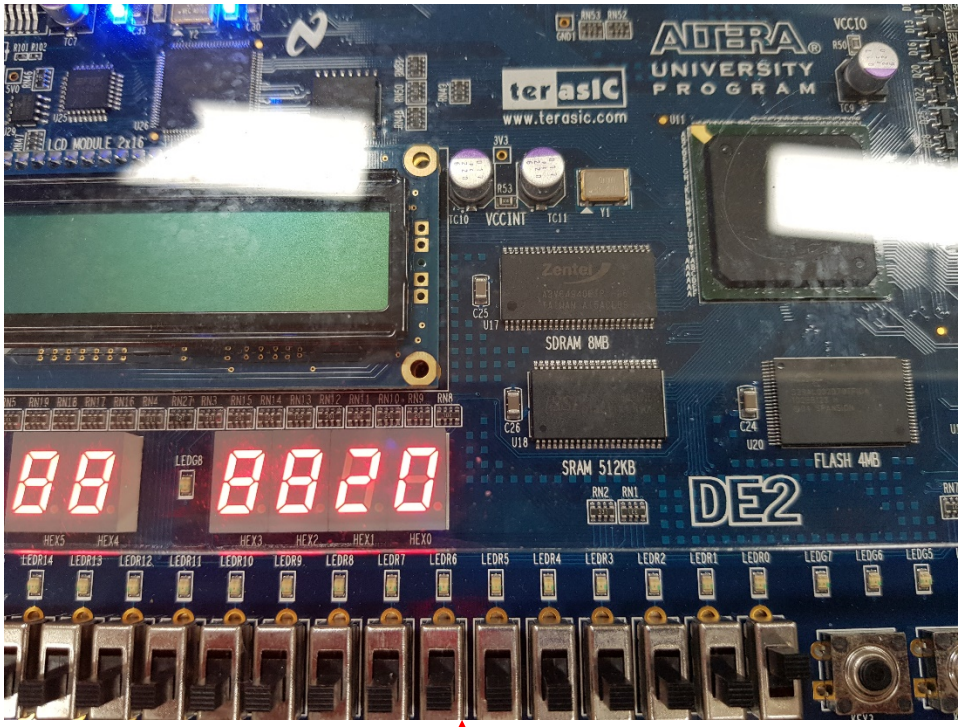


- **Stopwatch Secondary Block Port Mapping**

	Partition Hierarchy	root_partition	Yes
pin_name	Location	PIN_N2	Yes
q[25]	Location	PIN_Y18	Yes
seg0[0]	Location	PIN_AF10	Yes
seg0[1]	Location	PIN_AB12	Yes
seg0[2]	Location	PIN_AC12	Yes
seg0[3]	Location	PIN_AD11	Yes
seg0[4]	Location	PIN_AE11	Yes
seg0[5]	Location	PIN_V14	Yes
seg0[6]	Location	PIN_V13	Yes
sw_01	Location	PIN_N25	Yes
seg1[0]	Location	PIN_V20	Yes
seg1[1]	Location	PIN_V21	Yes
seg1[2]	Location	PIN_W21	Yes
seg1[3]	Location	PIN_Y22	Yes
seg1[4]	Location	PIN_AA24	Yes
seg1[5]	Location	PIN_AA23	Yes
seg1[6]	Location	PIN_AB24	Yes

➢ Result of Stopwatch secondary block





If 0-9 is counted, the decimal segment is additionally marked
Therefore, a two-digit decimal stopwatch display was possible.

Thank you