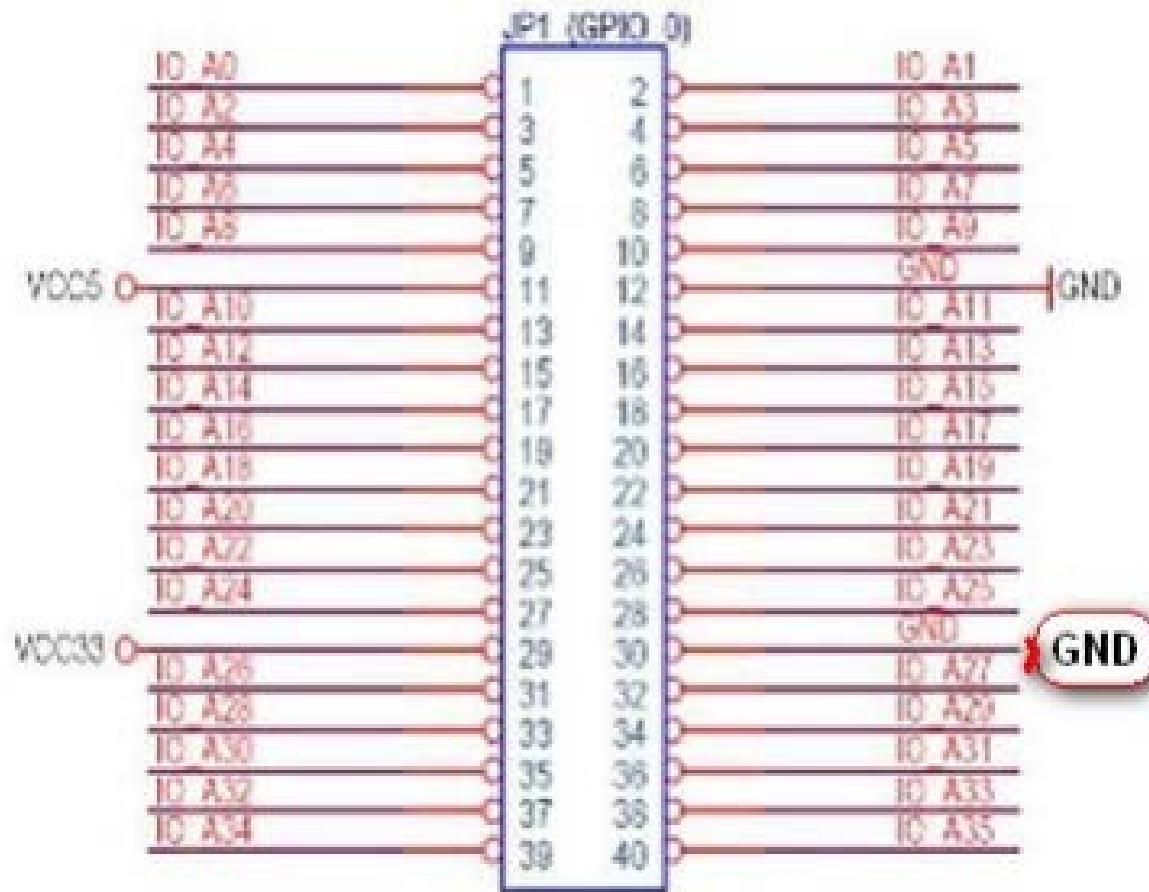


Adc-dac 101

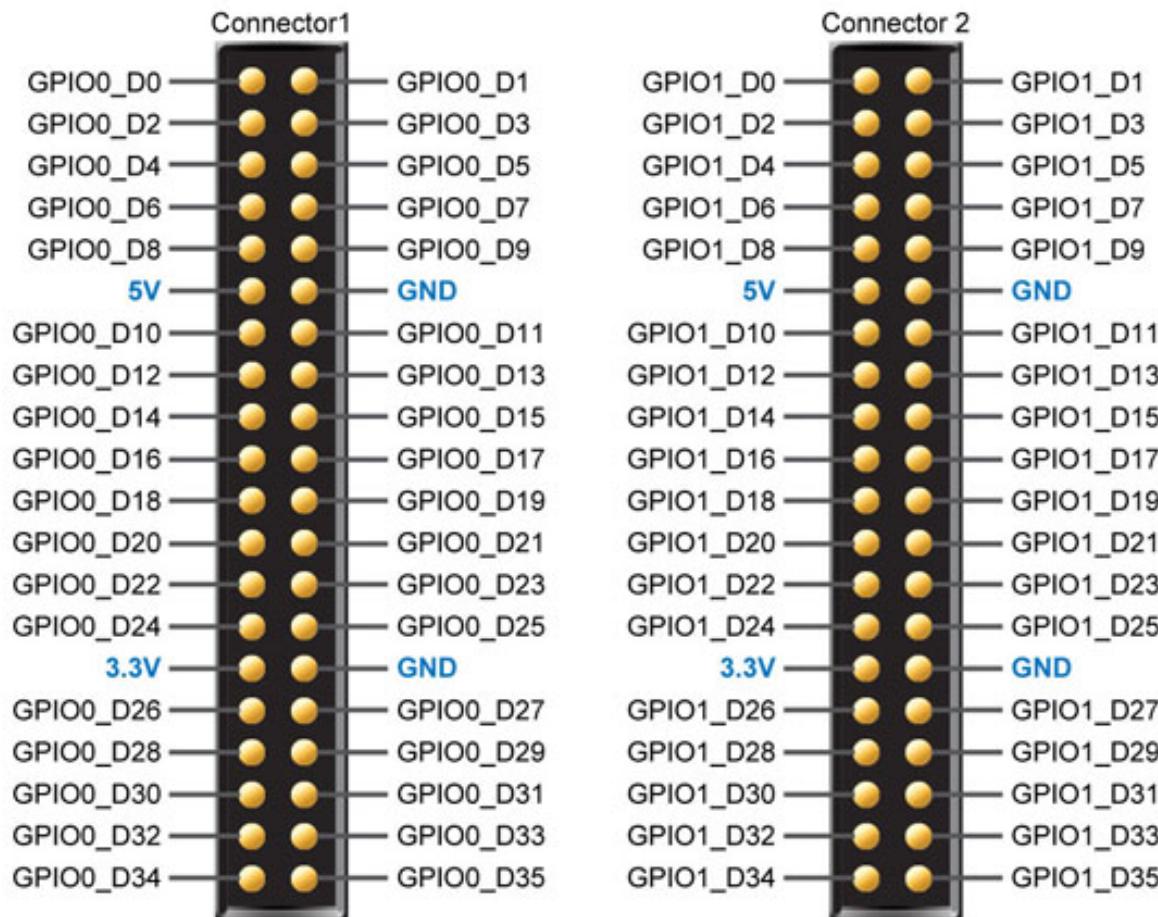
강원대학교 일반대학원
전자공학전공 석사 박제창

➤ DE2 보드의 확장 gpio 포트

	A	B
361	GPIO_0[0]	PIN_D25
362	GPIO_0[1]	PIN_J22
363	GPIO_0[2]	PIN_E26
364	GPIO_0[3]	PIN_E25
365	GPIO_0[4]	PIN_F24
366	GPIO_0[5]	PIN_F23
367	GPIO_0[6]	PIN_J21
368	GPIO_0[7]	PIN_J20
369	GPIO_0[8]	PIN_F25
370	GPIO_0[9]	PIN_F26
371	GPIO_0[10]	PIN_N18
372	GPIO_0[11]	PIN_P18
373	GPIO_0[12]	PIN_G23
374	GPIO_0[13]	PIN_G24
375	GPIO_0[14]	PIN_K22
376	GPIO_0[15]	PIN_G25
377	GPIO_0[16]	PIN_H23
378	GPIO_0[17]	PIN_H24
379	GPIO_0[18]	PIN_J23
380	GPIO_0[19]	PIN_J24
381	GPIO_0[20]	PIN_H25
382	GPIO_0[21]	PIN_H26
383	GPIO_0[22]	PIN_H19
384	GPIO_0[23]	PIN_K18
385	GPIO_0[24]	PIN_K19
386	GPIO_0[25]	PIN_K21
387	GPIO_0[26]	PIN_K23
388	GPIO_0[27]	PIN_K24
389	GPIO_0[28]	PIN_L21
390	GPIO_0[29]	PIN_L20
391	GPIO_0[30]	PIN_J25
392	GPIO_0[31]	PIN_J26
393	GPIO_0[32]	PIN_L23
394	GPIO_0[33]	PIN_L24

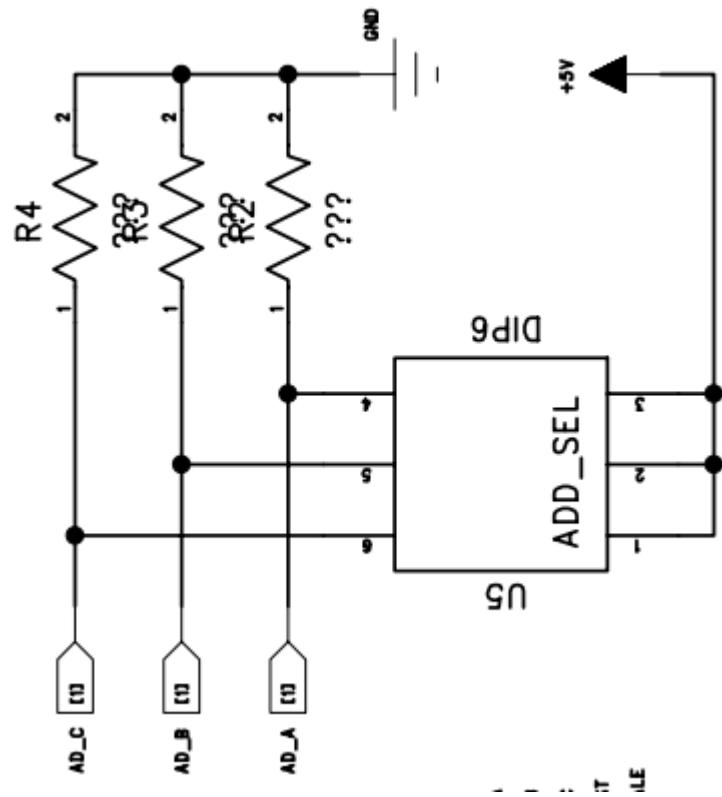


➤ DE2 보드의 확장 gpio 포트



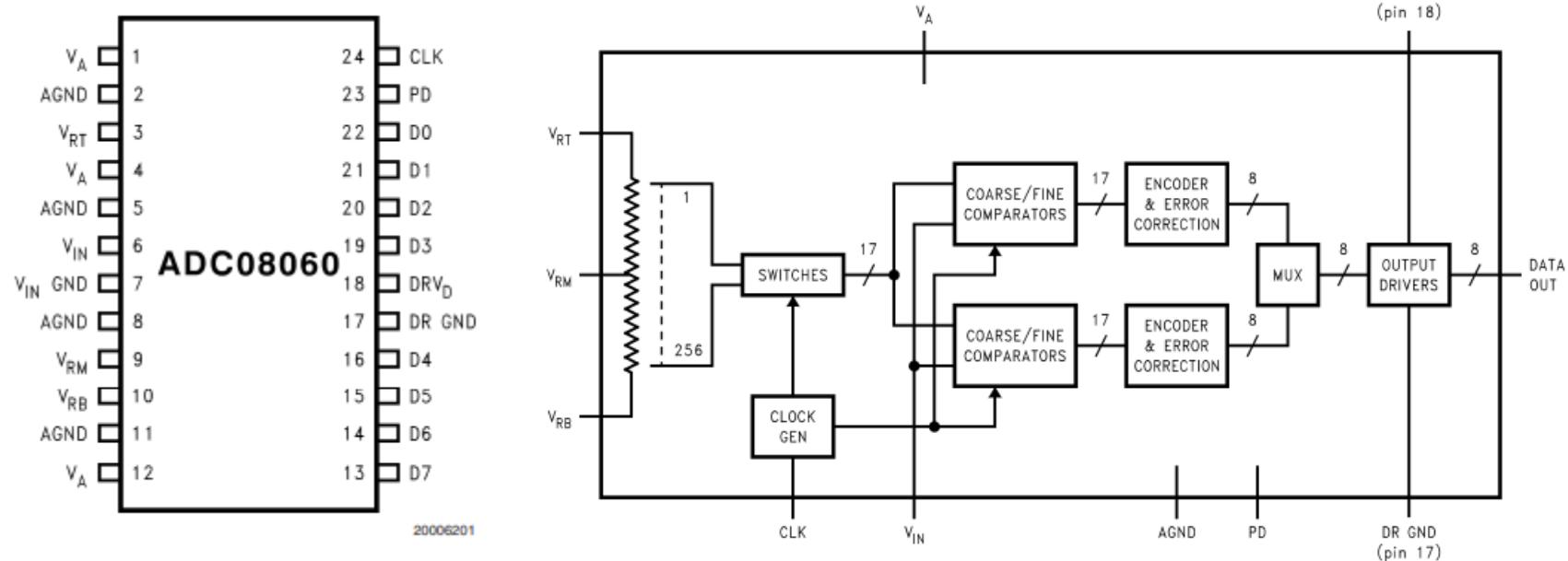
➤ ADC0806 변환 포트 설정 부분

The ADC0806 is a low-power, 8-bit, monolithic analog-to-digital converter with an on-chip track-and-hold circuit. Optimized for low cost, low power, small size and ease of use, this product operates at conversion rates of 20 MSPS to 70 MSPS with outstanding dynamic performance over its full operating range while consuming just 1.3 mW per MHz of clock frequency. That's just 78 mW of power at 60 MSPS. Raising the PD pin puts the ADC0806 into a Power Down mode where it consumes just 1 mW.



➤ ADC08060

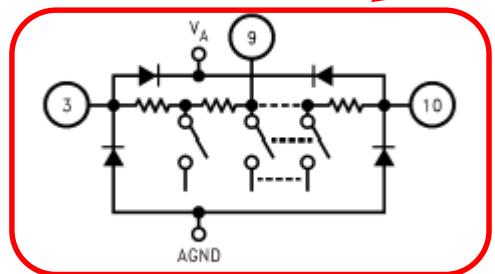
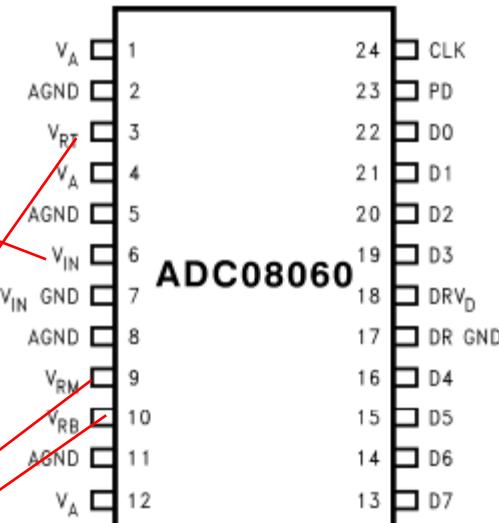
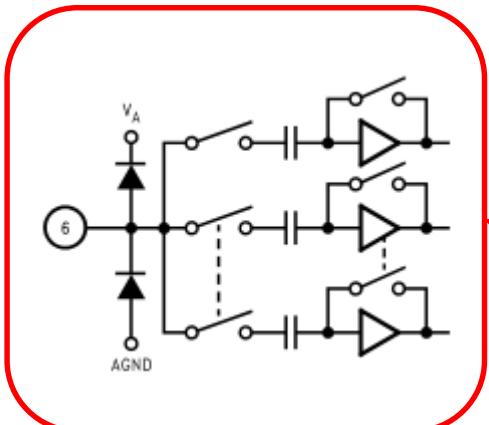
ADC08060 8-Bit, 20 MSPS to 60 MSPS, 1.3 mW/MSPS A/D Converter



20006201

➤ ADC08060

ADC08060 8-Bit, 20 MSPS to 60 MSPS, 1.3 mW/MSPS A/D Converter



Converter Electrical Characteristics

The following specifications apply for $V_A = DR$, $V_D = +3.0V_{DC}$, $V_{RT} = +1.9V$, $V_{RB} = 0.3V$, $C_L = 10\text{ pF}$, $f_{CLK} = 60\text{ MHz}$ at 50% duty cycle. **Boldface limits apply for $T_A = T_{MIN}$ to T_{MAX}** ; all other limits $T_A = 25^\circ C$ (Notes 7, 8)

Symbol	Parameter	Conditions	Typical (Note 9)	Limits (Note 9)	Units (Limits)
DC ACCURACY					
INL	Integral Non-Linearity		± 0.5	± 1.3	LSB (max)
DNL	Differential Non-Linearity		± 0.4	$+1.0$ -0.9	LSB (max) LSB (min)
	Missing Codes			0	(max)
FSE	Full Scale Error		18	± 28	mV (max)
ZSE	Zero Scale Offset Error		26	± 35	mV (max)

➤ ADC08060

ADC08060 8-Bit, 20 MSPS to 60 MSPS, 1.3 mW/MSPS A/D Converter

Timing Diagram

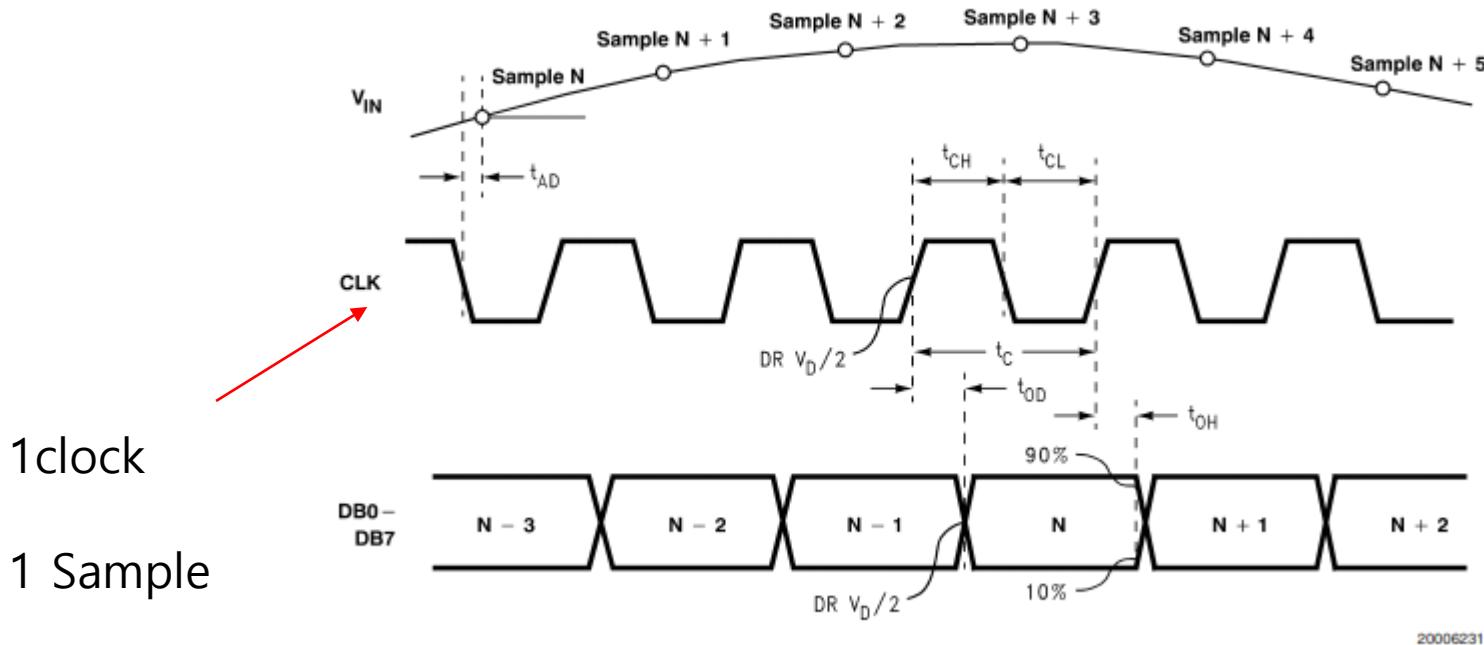
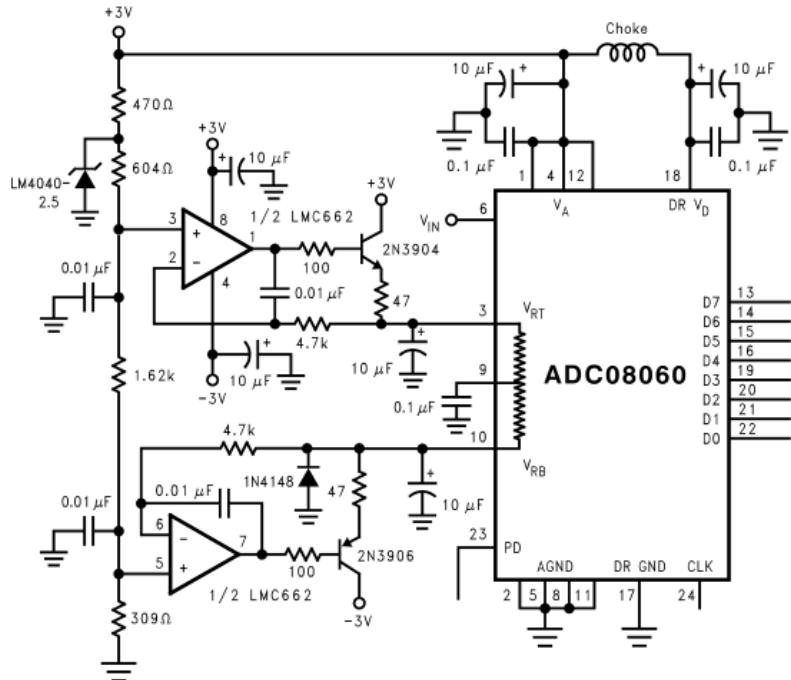


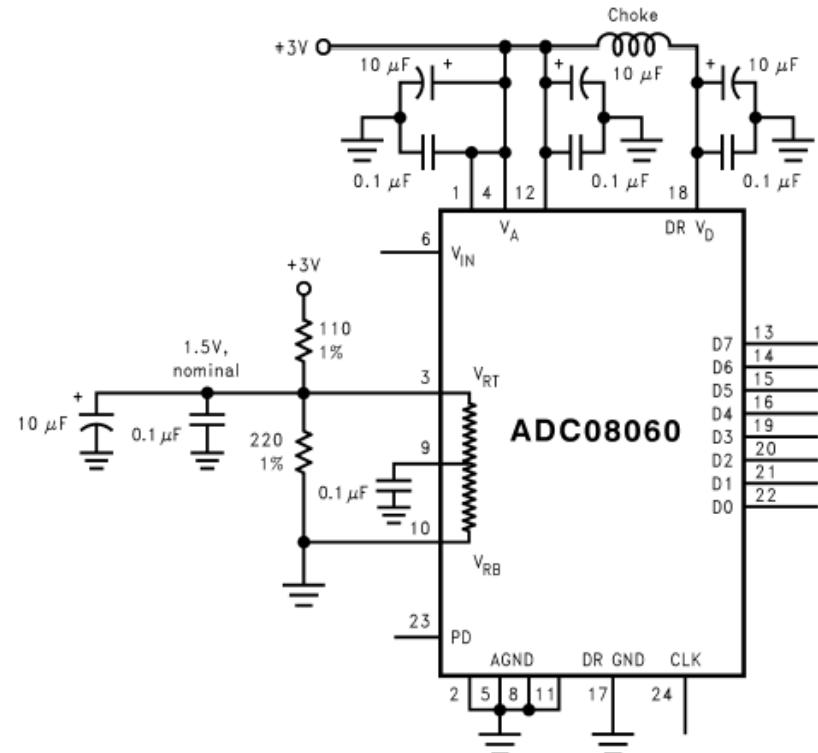
FIGURE 1. ADC08060 Timing Diagram

➤ ADC08060

ADC08060 8-Bit, 20 MSPS to 60 MSPS, 1.3 mW/MSPS A/D Converter



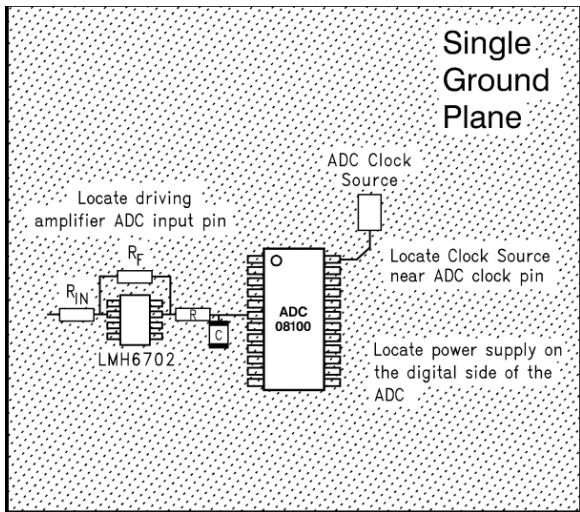
20006233



20006232

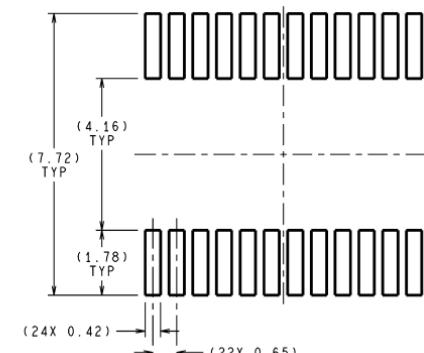
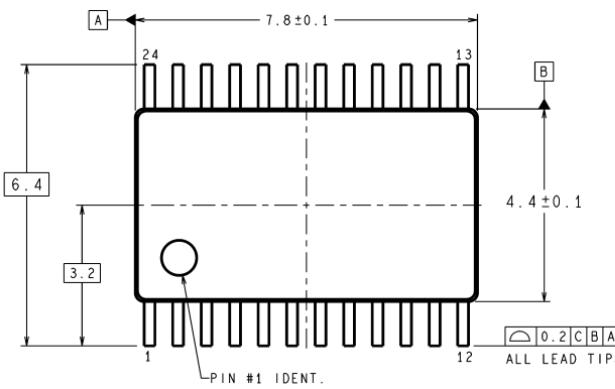
➤ ADC08060

ADC08060 8-Bit, 20 MSPS to 60 MSPS, 1.3 mW/MSPS A/D Converter

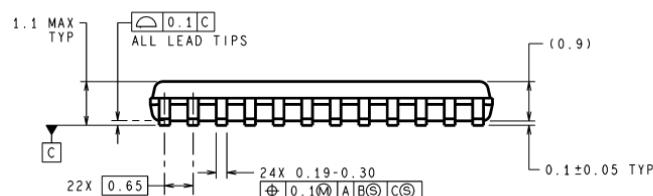


PCB Pattern

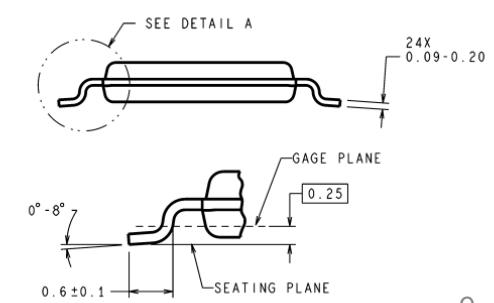
PCB Spec



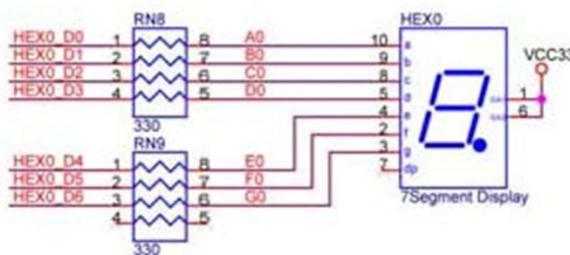
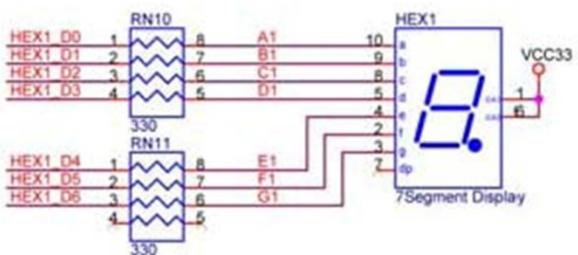
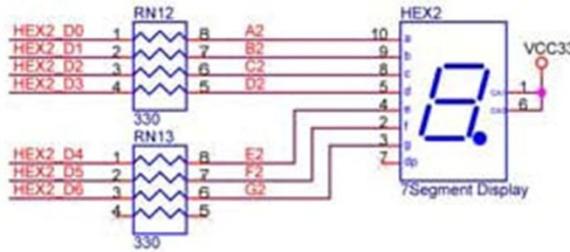
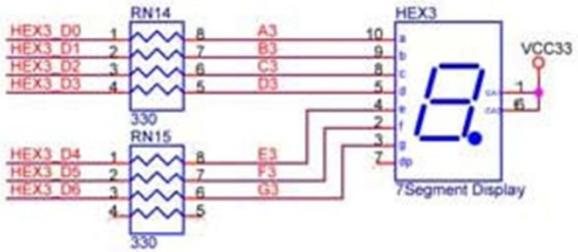
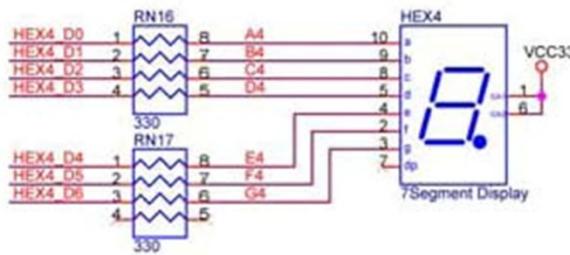
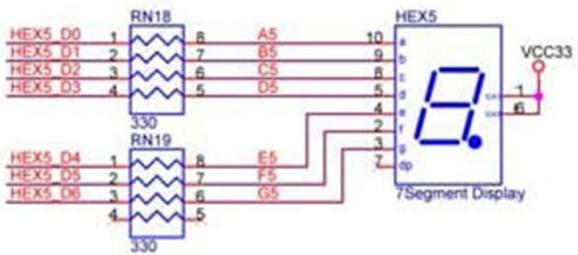
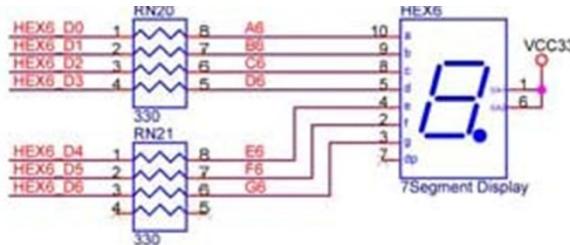
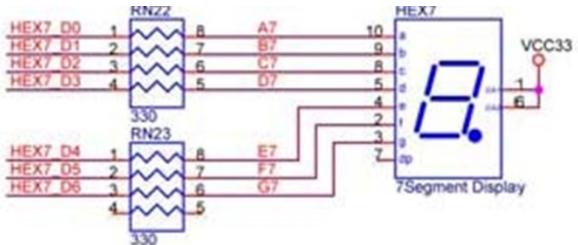
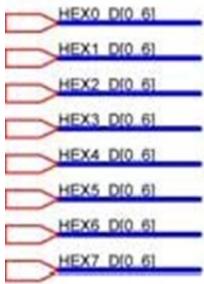
LAND PATTERN RECOMMENDATION



DIMENSIONS ARE IN MILLIMETERS



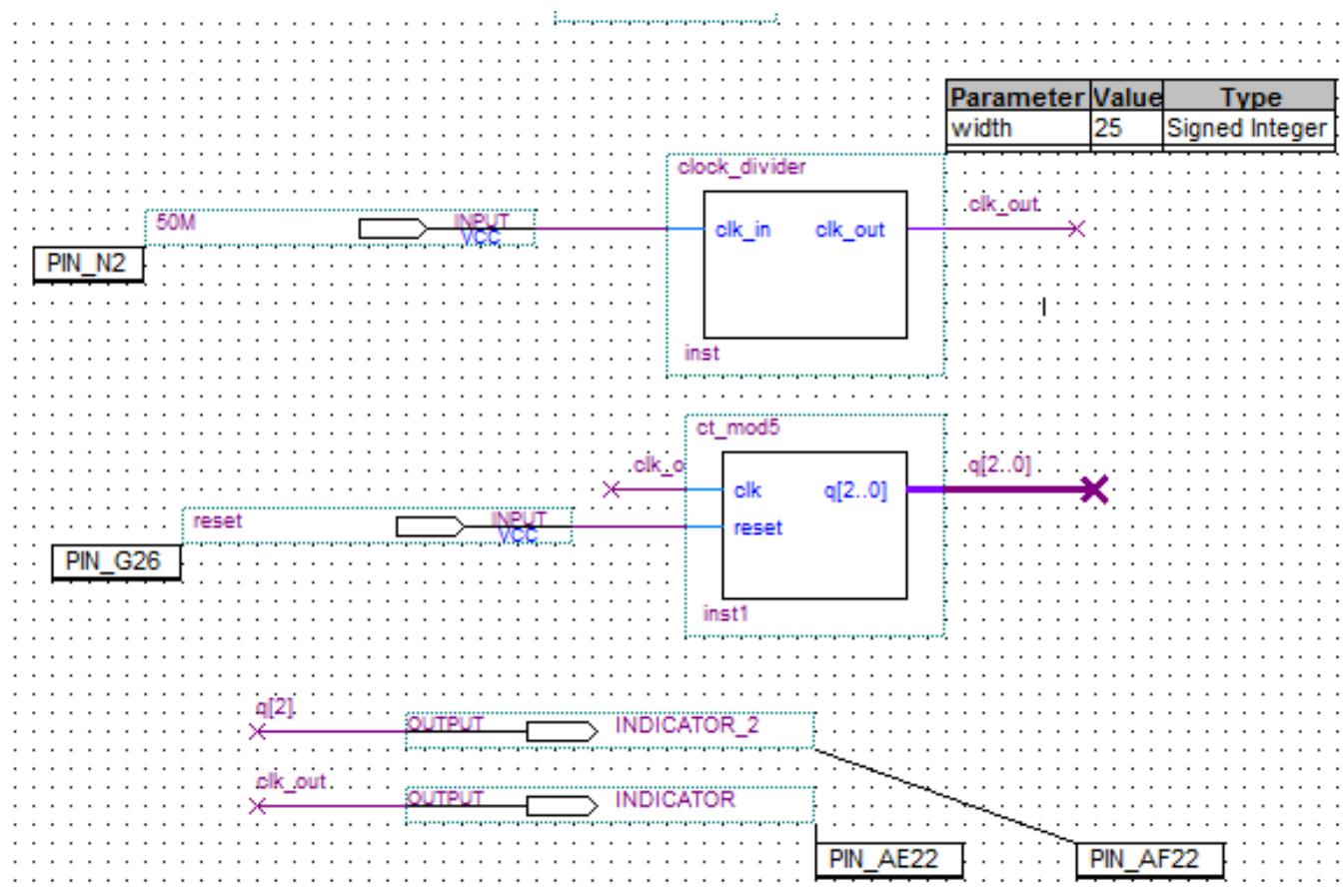
DETAIL A
TYPICAL, SCALE:20X



- Clock division
- Verilog
- 50MHz ➔ 1hz

```
1  module PLL_1Hz(
2      input wire clk_50MHz,
3      output reg clk_1Hz
4  );
5
6      reg [31:0]counter;
7
8      always@(posedge clk_50MHz)
9      begin
10         if(counter==32'd24999999)
11             begin
12                 clk_1Hz <= ~clk_1Hz;
13                 counter <= 32'b0;
14             end
15             else counter <= counter + 32'b1;
16         end
17
18     endmodule
19
```

➤ Adc 변환 클럭 분주 회로



카운터를 통해 클럭 분주하기

The scaling factor

The frequency divider is a simple component which objective is to reduce the input frequency. The component is implemented through the use of the scaling factor and a counter. The scaling factor is the relation between the input frequency and the desired output frequency:

$$Scale = \frac{f_{in}}{f_{out}}$$

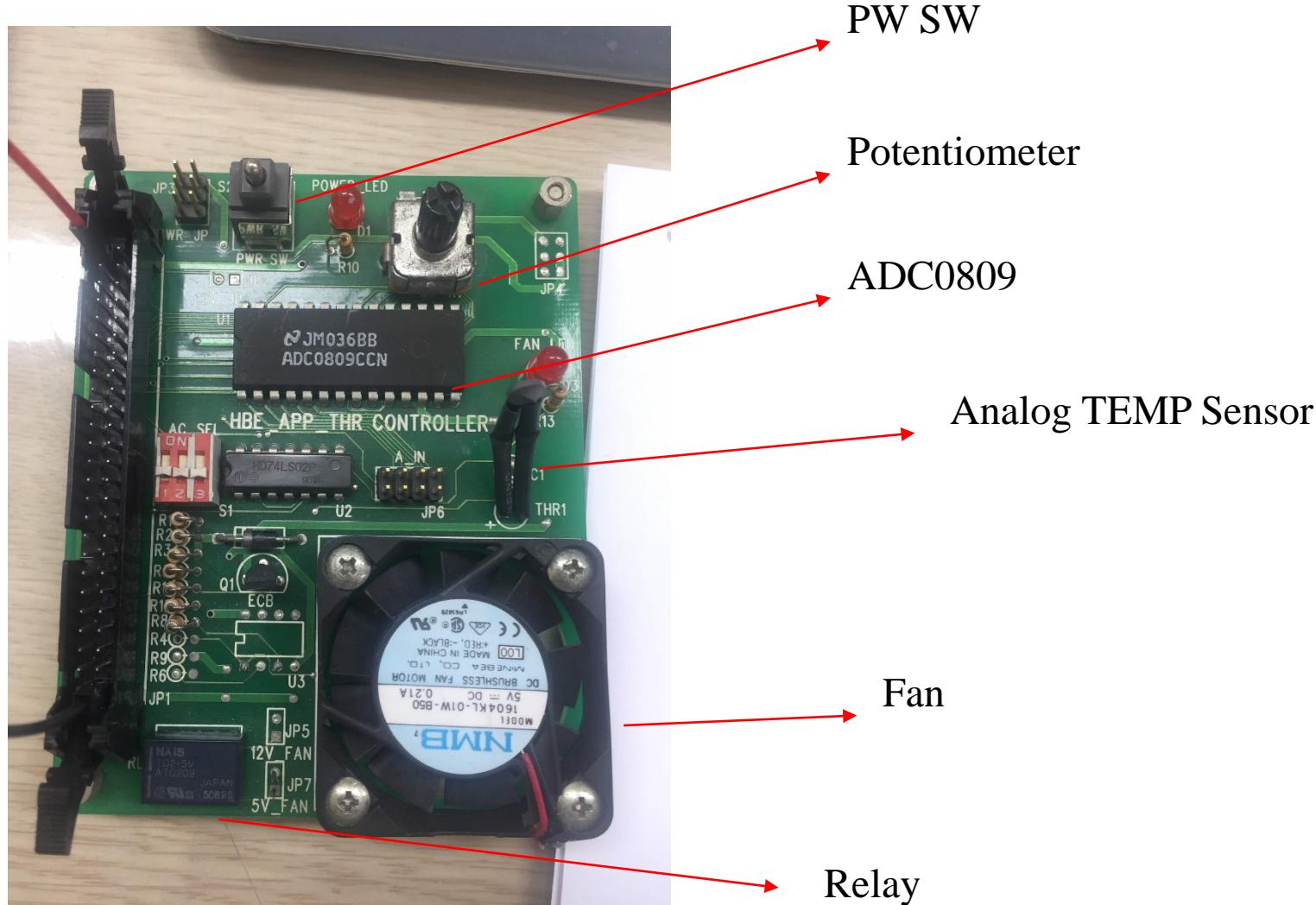
Assuming an input frequency of 50MHz and provided we need an output frequency of 200Hz, we yield:

$$Scale = \frac{50MHz}{200Hz} = 250000$$

Therefore, the counter of the frequency divider generates the output signal of 200Hz each 250000 cycles.

한백 전자의 adc-dac 보드를 활용한다.

ADC는 adc0809로 8비트 해상도를 가지며 256까지 아날로그 입력 신호를 디지털 신호로 변환 가능하다.



➤ ADC0808/ADC0809 8-Bit µP Compatible A/D Converters with 8-Channel Multiplexer

FEATURES

- **Easy Interface to All Microprocessors**
- **Operates Ratiometrically or with 5 V_{DC} or Analog Span Adjusted Voltage Reference**
- **No Zero or Full-Scale Adjust Required**
- **8-Channel Multiplexer with Address Logic**
- **0V to V_{CC} Input Range**
- **Outputs meet TTL Voltage Level Specification**
- **ADC0808 Equivalent to MM74C949**
- **ADC0809 Equivalent to MM74C949-1**

KEY SPECIFICATIONS

- **Resolution: 8 Bits**
- **Total Unadjusted Error: ±½ LSB and ±1 LSB**
- **Single Supply: 5 VDC**
- **Low Power: 15 mW**
- **Conversion Time: 100 µs**

The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8-single-ended analog signals.

The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE outputs.

➤ ADC0808/ADC0809 8-Bit µP Compatible A/D Converters with 8-Channel Multiplexer

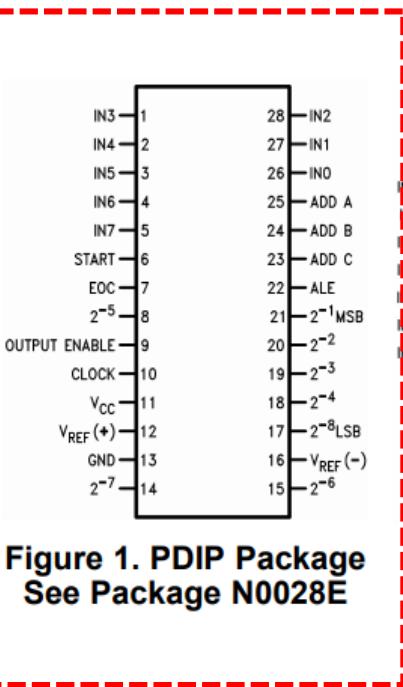


Figure 1. PDIP Package
See Package N0028E

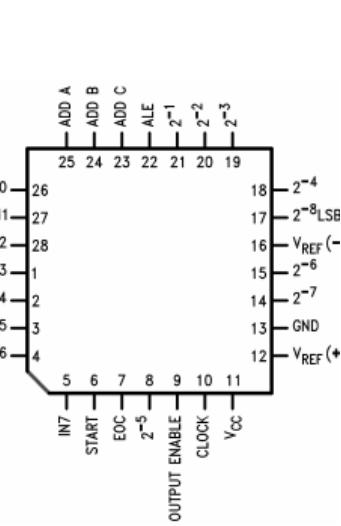
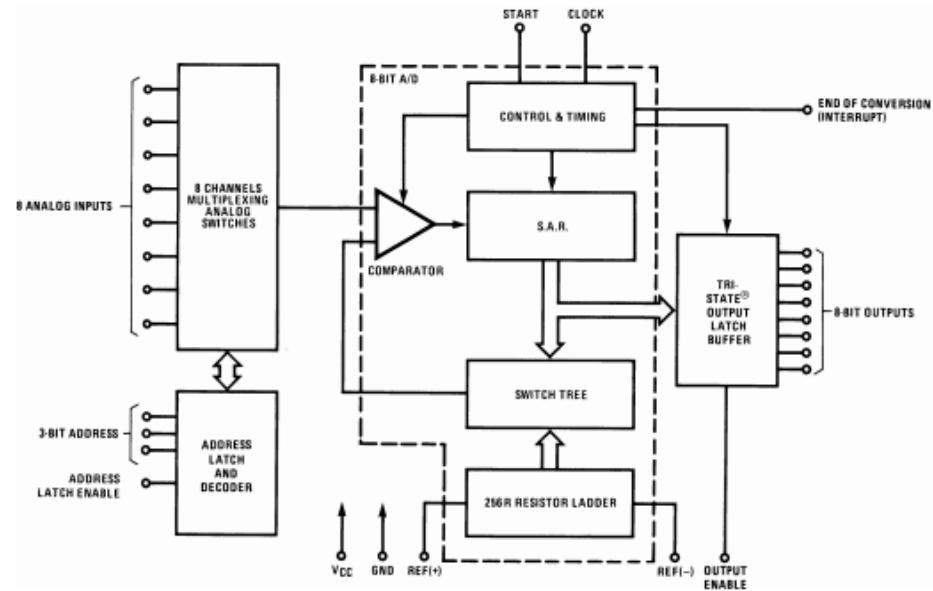


Figure 2. PLCC
Package
See Package FN0028A

Block Diagram



➤ ADC0808/ADC0809 8-Bit µP Compatible A/D Converters with 8-Channel Multiplexer

MULTIPLEXER

The device contains an 8-channel single-ended analog signal multiplexer. A particular input channel is selected by using the address decoder. [Table 1](#) shows the input states for the address lines to select any channel. The address is latched into the decoder on the low-to-high transition of the address latch enable signal.

Table 1. Analog Channel Selection

SELECTED ANALOG CHANNEL	ADDRESS LINE		
	C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

➤ ADC0808/ADC0809 8-Bit μ P Compatible A/D Converters with 8-Channel Multiplexer

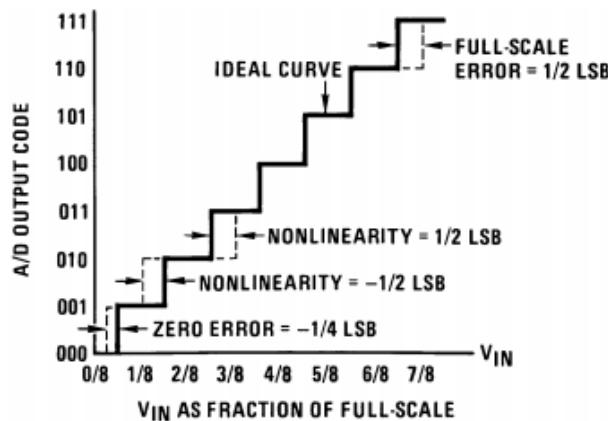


Figure 4. 3-Bit A/D Transfer Curve

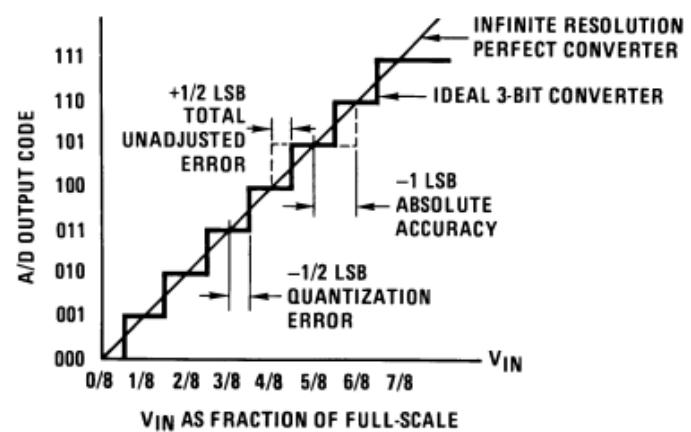
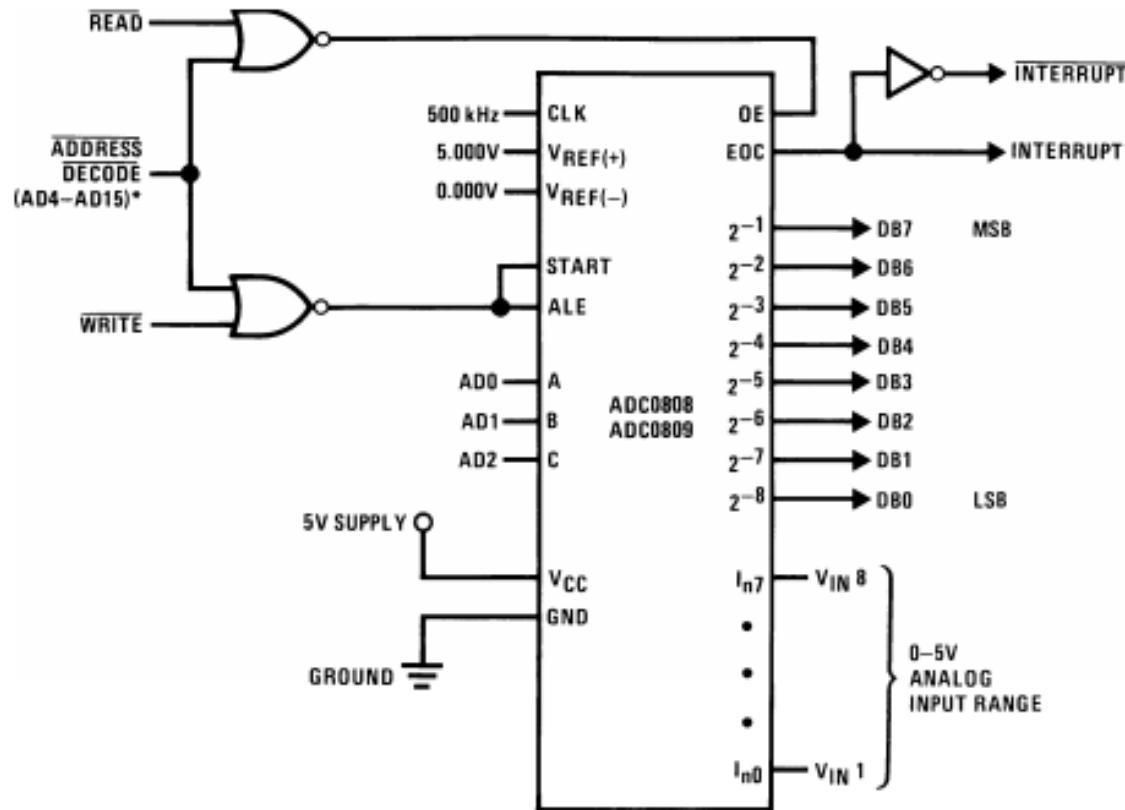


Figure 5. 3-Bit A/D Absolute Accuracy Curve

CLOCK은 ADC0809에 동작 클럭을 공급하는 핀이다. 사용할 수 있는 클록 주파수는 640kHz가 표준이며 최대 1280kHz까지 지원한다.

➤ ADC0808/ADC0809 8-Bit µP Compatible A/D Converters with 8-Channel Multiplexer



➤ ADC0808/ADC0809 8-Bit μP Compatible A/D Converters with 8-Channel Multiplexer

ADC0809의 핀 기능에 대해 간단히 설명하면, VIN(+)와 VIN(-)는 아날로그 입력 핀이다.. 차동 입력으로 되어 있어서, VIN(+) ~ VIN(-) 전압값이 디지털 값으로 변환된다. 입력 전압 범위 0~5V이다.

① D00~D07은 디지털 출력 핀입니다.

변환된 디지털 데이터를 8비트 2진값(D00가 LSB)로 출력된다. 입력이 VIN(+) ~ VIN(-) = 0V일 때 디지털 데이터 값은 "00000000"(10진수 0)으로, 5V일 때는 "11111111"(10진수 255)으로 출력된다. 그 중간의 입력 전압은 이것에 비례한 출력값이 된다. 이 D00~D07은 3스테이트 출력 구조로 구성되어 있다.

② SEL_A, SEL_B, SEL_C는 8 개의 아날로그 입력 데이터 라인을 선택하는 제어 명령으로 사용하는 입력핀이다. ADC0809에서는 총 8개의 아날로그 입력 데이터 라인 (ANA_IN0 ~ ANA_IN7)이 있는데, SEL_A, SEL_B, SEL_C 값이 모두 low('0')일 때 첫 번째 데이터 라인(ANA_IN0)을 선택하며, 모든 값이 high('1')일 때 마지막 데이터 라인(ANA_IN7)을 선택하여 이 데이터 라인을 통해 들어오는 아날로그 입력 값을 변환시킨다.

➤ ADC0808/ADC0809 8-Bit μP Compatible A/D Converters with 8-Channel Multiplexer

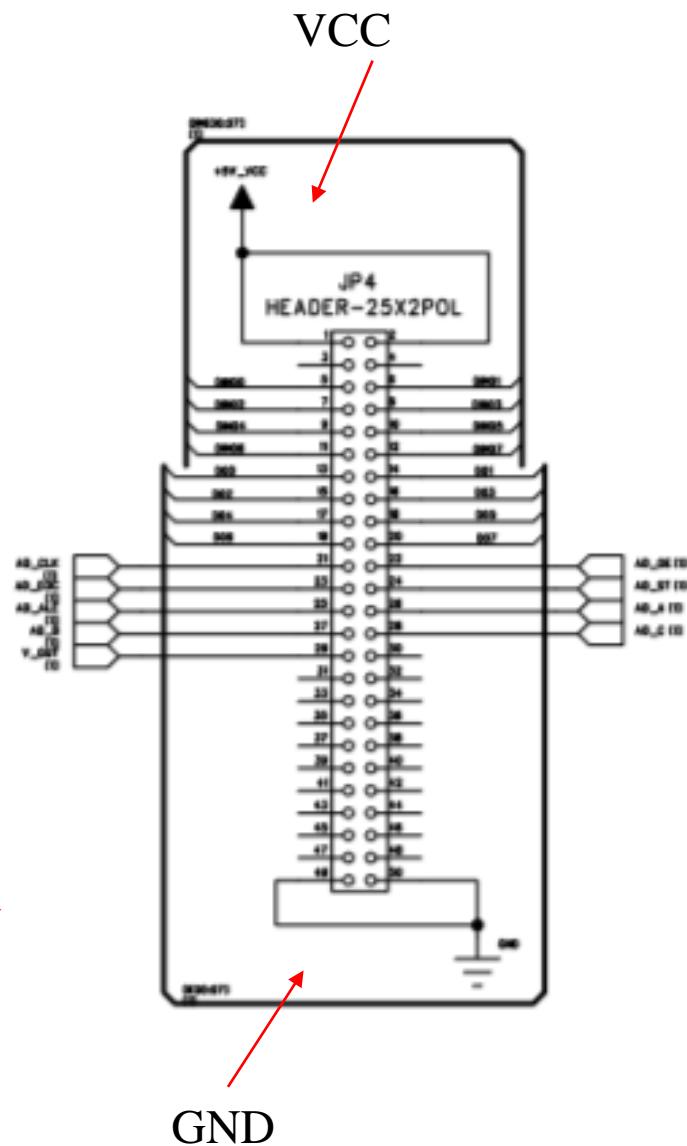
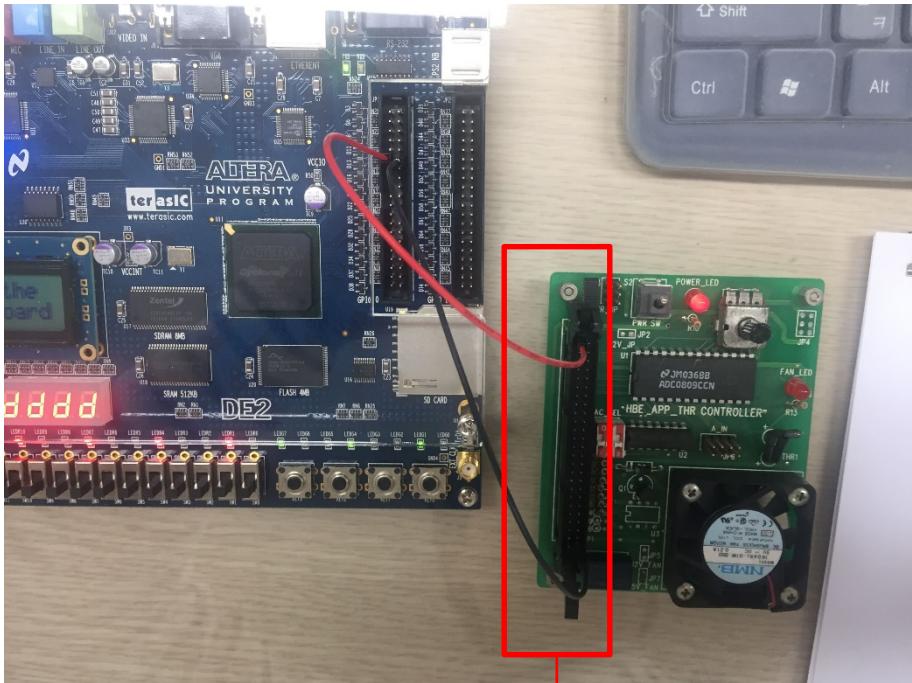
③ /CS, /RD, /WR은 제어 명령으로 받을 수 있는 입력 핀이다.

ADC0809는 /CS와 /WR이 'Low'로될 때 변환 동작이 시작되고, 동작이 시작된 후에 /CS와 /RD가 'Low'로 되면 변환 결과를 D00~D07에 출력한다.

④ EOC는 변환 동작을 완료했을 때에 액티브되는 출력 핀이다.

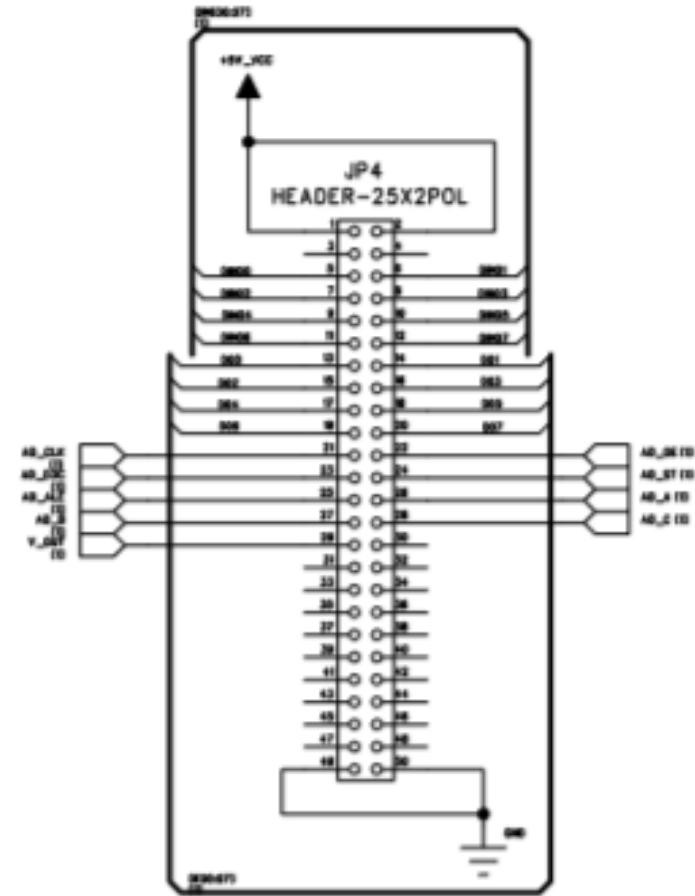
일반적인 A/D 컨버터에서는 변환 시작 명령을 받고서, 변환이 완료될 때까지 어느 정도의 시간이 걸린다. ADC0809의 경우에는 약 100us 시간이 걸리는데, 변환 전에 데이터를 읽으면 그 데이터는 올바르게 인식하지 않기 때문에 그 변환 시간만큼을 지연시켜야 한다. EOC를 인터럽트 기능으로 사용한다면 변환 완료 시점에서 인터럽트를 요청할 수 있다. 아래 [그림]에 이 신호의 타이밍 도가 표시되어 있다.

➤ Connected De2 with Hanback ADCDAC



➤ Hanback ADCDAC

핀 기능	J2 핀 번호		핀 기능
+ 5V	1	2	+ 5V
AIN00	3	4	AIN01
AIN02	5	6	AIN03
AIN04	7	8	AIN05
AIN06	9	10	AIN07
V_OUT	11	12	L_OUT
GND	13	14	GND

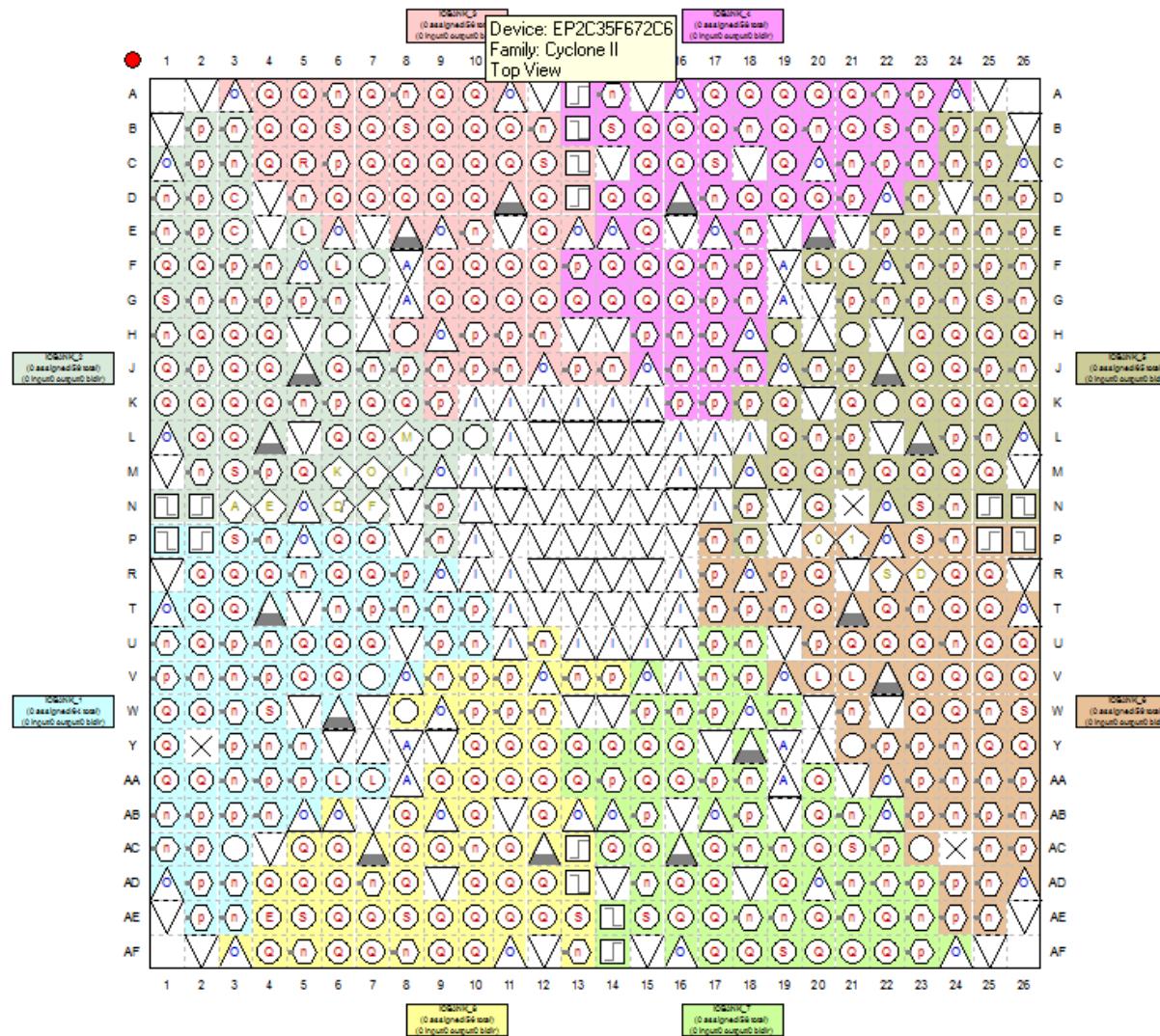


➤ Hanback ADCDAC

확장 모듈 PCB 실제 이름 및 핀 번호		HBE-DTK-COMBO 보드로 연결할 때의 포트 및 핀 번호		HBE-DTK-COMBO 보드로 연결할 때의 포트 및 핀 번호	
핀 이름	EPF1KXXQC208	EPF1KXXQC208	포트 및 핀 번호	XC2SXXPQ208	포트 및 핀 번호
VCC (+ 5V)	VCC (+ 5V)	VCC (+ 5V)	JP4-1	VCC (+ 5V)	EXT3-1
VCC (+ 5V)	VCC (+ 5V)	VCC (+ 5V)	JP4-2	VCC (+ 5V)	EXT3-2
N.C.			JP4-3		EXT3-3
N.C.			JP4-4		EXT3-4
DIN00	16	16	JP4-5	176	EXT3-5
DIN01	15	15	JP4-6	176	EXT3-6
DIN02	14	14	JP4-7	174	EXT3-7
DIN03	13	13	JP4-8	173	EXT3-8
DIN04	12	12	JP4-9	172	EXT3-9
DIN05	11	11	JP4-10	168	EXT3-10
DIN06	10	10	JP4-11	167	EXT3-11
DIN07	9	9	JP4-12	166	EXT3-12
D00	8	8	JP4-13	165	EXT3-13
D01	7	7	JP4-14	164	EXT3-14
D02	208	208	JP4-15	163	EXT3-15
D03	207	207	JP4-16	162	EXT3-16
D04	206	206	JP4-17	161	EXT3-17
D05	205	205	JP4-18	160	EXT3-18
D06	81	81	JP4-19	193	EXT3-19
D07	80	80	JP4-20	192	EXT3-20
AD_CLK	29	29	JP4-21	191	EXT3-21
AD_OE	28	28	JP4-22	189	EXT3-22
AD_EOC	27	27	JP4-23	188	EXT3-23
AD_ST	26	26	JP4-24	187	EXT3-24
AD_ALE	25	25	JP4-25	181	EXT3-25
AD_A	24	24	JP4-26	180	EXT3-26
AD_B	92	92	JP4-27	41	EXT3-27
AD_C	90	90	JP4-28	37	EXT3-28
V_OUT	89	89	JP4-29	36	EXT3-29
N.C.	8 Bit AD/DA Converter Module에서는 30 ~ 38번 핀을 사용하지 않습니다.				
GND			JP4-49		EXT3-49
GND			JP4-50		EXT3-50

➤ Pin Assignment

Top View - Wire Bond Cyclone II - EP2C35F672C6



➤ Mif 파일 만들기

- └ VHDL File
- └ Memory Files
 - └ Hexadecimal (Intel-Format) File
 - └ Memory Initialization File
- └ Verification/Debugging Files
 - └ In-System Sources and Probes File

Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0
56	0	0	0	0	0	0	0	0
64	0	0	0	0	0	0	0	0
72	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0
88	0	0	0	0	0	0	0	0
96	0	0	0	0	0	0	0	0
104	0	0	0	0	0	0	0	0
112	0	0	0	0	0	0	0	0
120	0	0	0	0	0	0	0	0
128	0	0	0	0	0	0	0	0
136	0	0	0	0	0	0	0	0
144	0	0	0	0	0	0	0	0
152	0	0	0	0	0	0	0	0
160	0	0	0	0	0	0	0	0
168	0	0	0	0	0	0	0	0
176	0	0	0	0	0	0	0	0
184	0	0	0	0	0	0	0	0
192	0	0	0	0	0	0	0	0
200	0	0	0	0	0	0	0	0
208	0	0	0	0	0	0	0	0
216	0	0	0	0	0	0	0	0
224	0	0	0	0	0	0	0	0
232	0	0	0	0	0	0	0	0
240	0	0	0	0	0	0	0	0
248	0	0	0	0	0	0	0	0

➤ Mif 파일 만들기

The screenshot shows the Quartus II interface with the following details:

- File Menu:** File, Edit, View, Project, Assignments, Processing, Tools, Window, Help.
- Project Navigator:** Entity (Cyclone BDF Im35).
- Utility Windows:** m35wow (selected), Full Screen Ctrl+Alt+Space, Im35wow.bdf, Compilation Report - Flow Summary, B122526.vhd.
- Cells Per Row:** A context menu is open, showing options: 1, 2, 4, 8 (selected), 16, 32, AutoFit.
- Table View:** Shows memory data in a grid. The columns are Addr, +0, +1, +2, +3, +4, +5, +6, +7. The rows range from 0 to f8. All cells contain the value 0.
- Bottom Navigation:** Hierarchy, Files, Design Units.
- Tasks:** Flow: Compilation. Task list:
 - Compile Design (Failed)
 - Analysis & Synthesis (Failed)
 - Fitter (Place & Route)
 - Assembler (Generate programming files)
 - Classic Timing Analysis
 - EDA Netlist Writer
 - Program Device (Open Programmer) (Success)

➤ Mif 파일 만들기

The screenshot shows the Quartus II interface. The top menu bar has 'View' selected. A dropdown menu under 'View' is open, showing options like 'Utility Windows', 'Cells Per Row' (which is currently set to 1), 'Address Radix', 'Memory Radix', and 'Show Delimiter Spaces'. To the right of the menu, there is a memory dump window titled 'Im35wow.bdf' showing memory addresses from 00 to 22 with values mostly set to 0.

Addr	Value
00	0
01	0
02	0
03	0
04	0
05	0
06	0
07	0
08	0
09	0
0a	0
0b	0
0c	0
0d	0
0e	0
0f	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
1a	0
1b	0
1c	0
1d	0
1e	0
1f	0
20	0
21	0
22	0

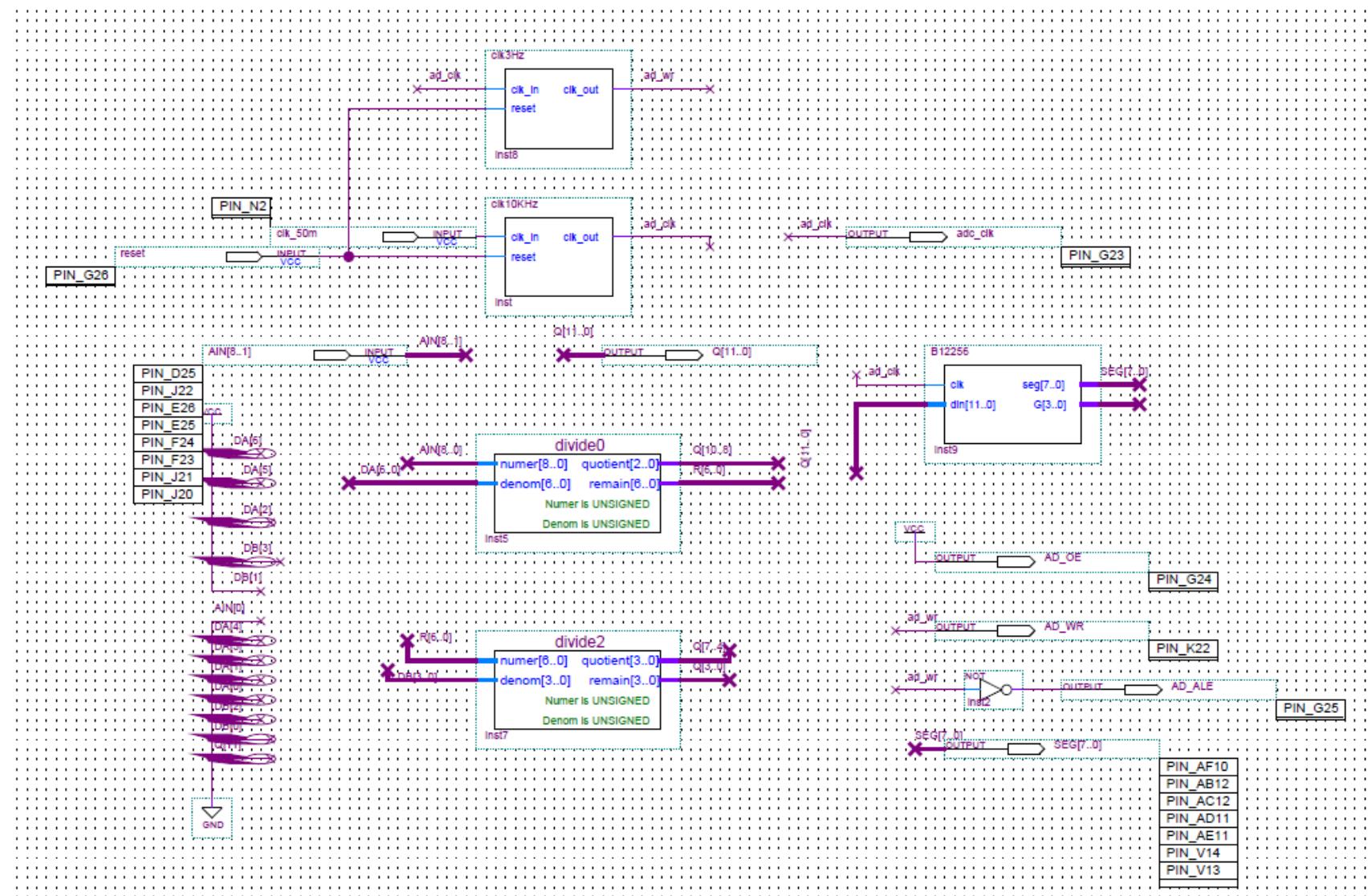
View 메뉴에 들어가서
셀 크기와 메모리 표기
방식을 사용자 입맛에
맞게 수정할 수 있다.

➤ 보드블록 설계도

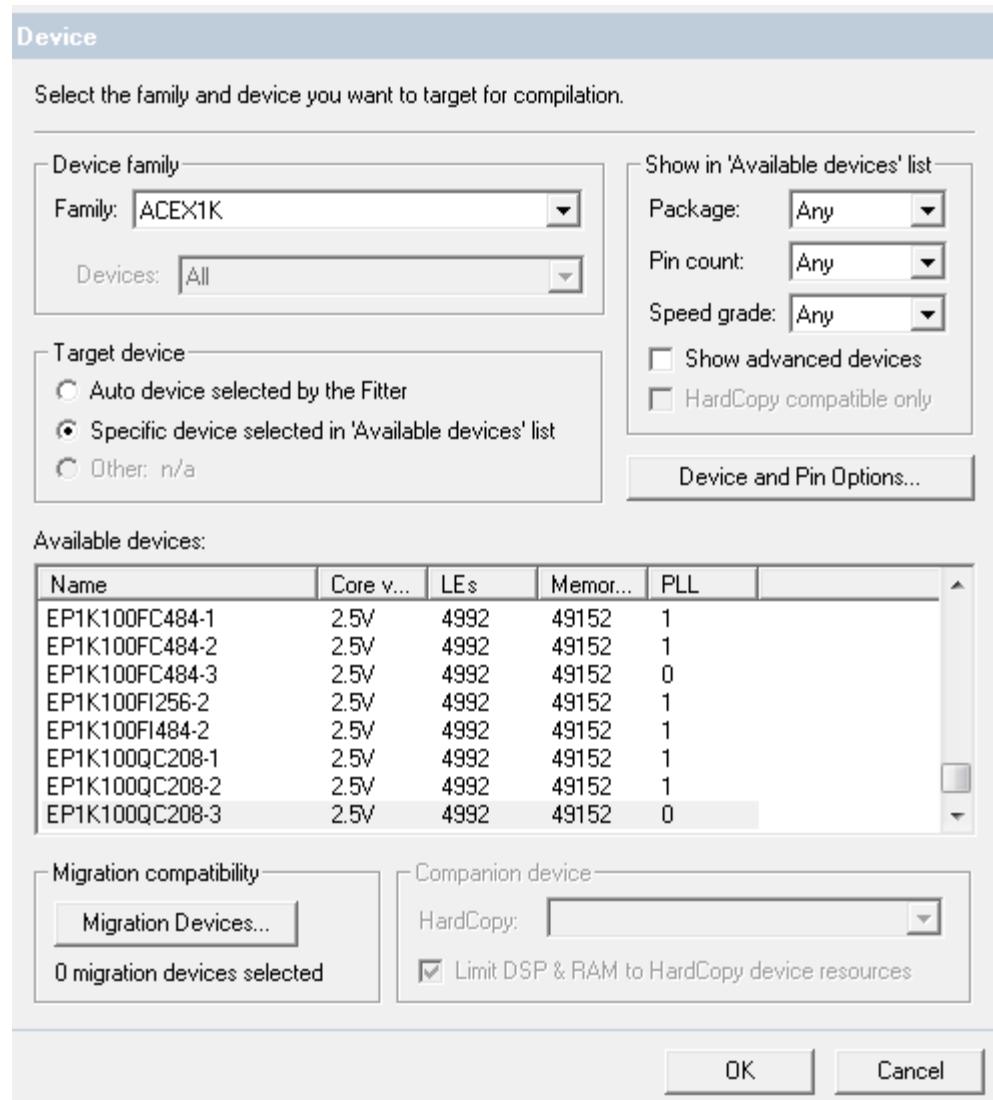
Date: November 21, 2018

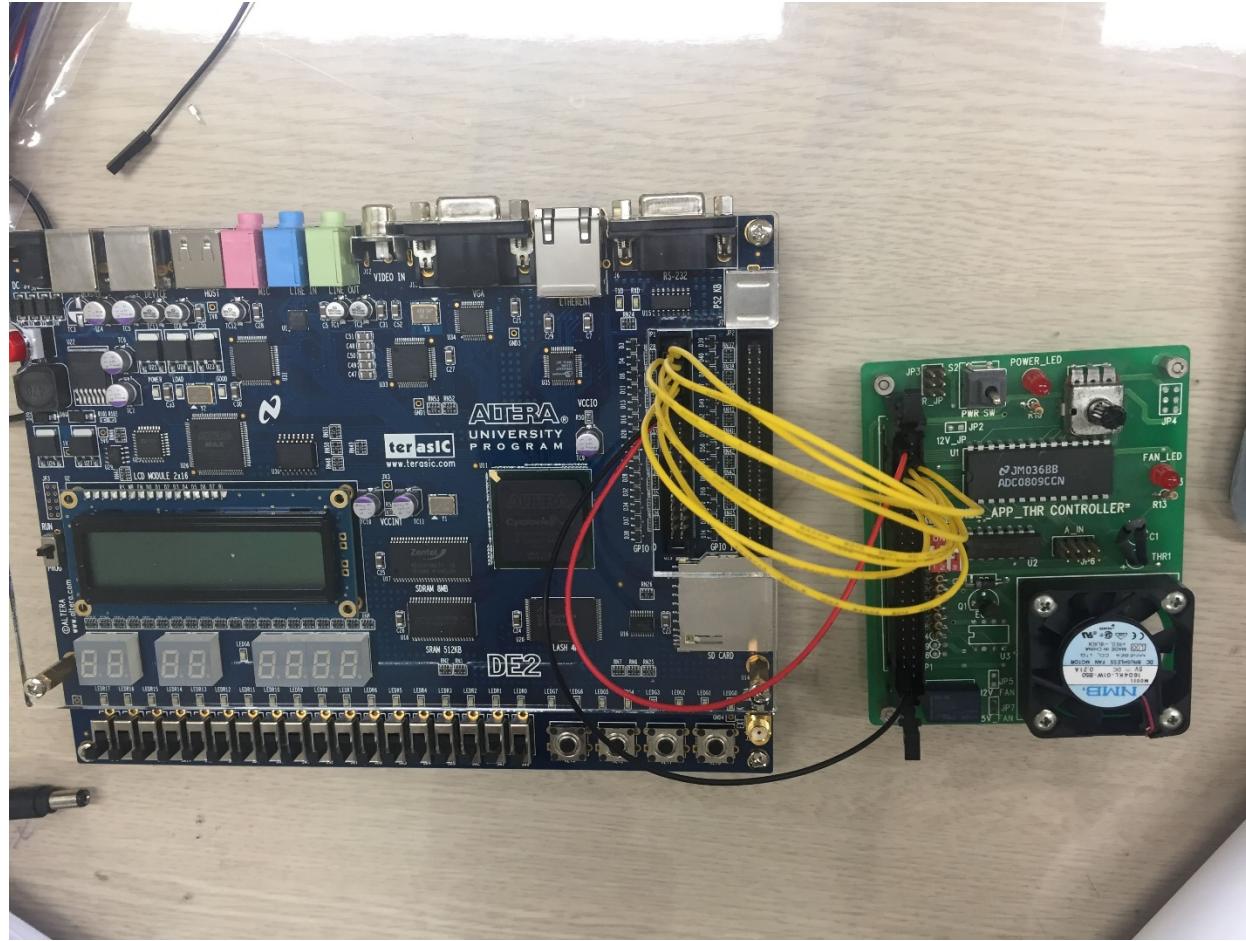
lm35wow.bdf*

Project: lm35wow

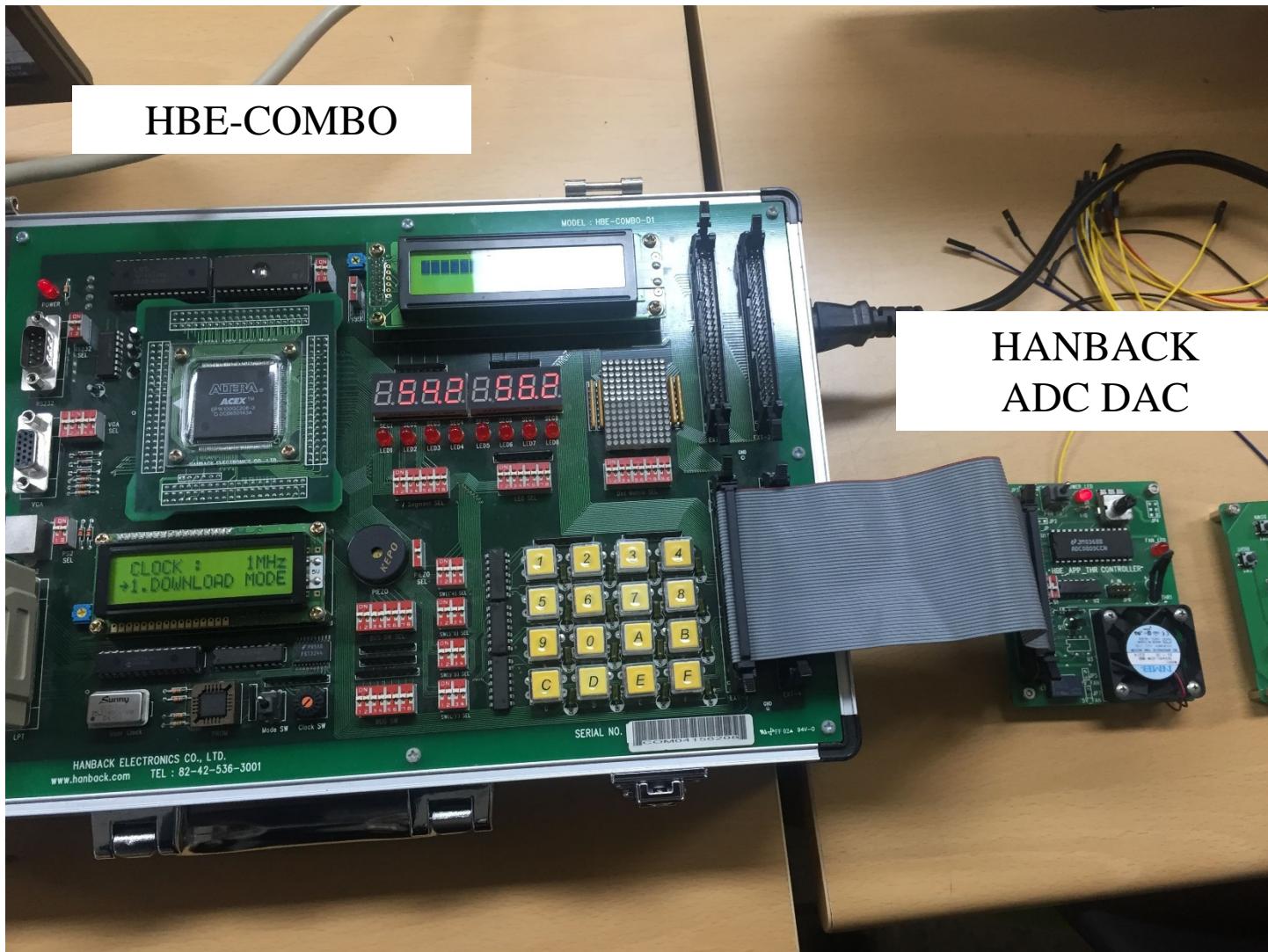


➤ 보드블록 설계도



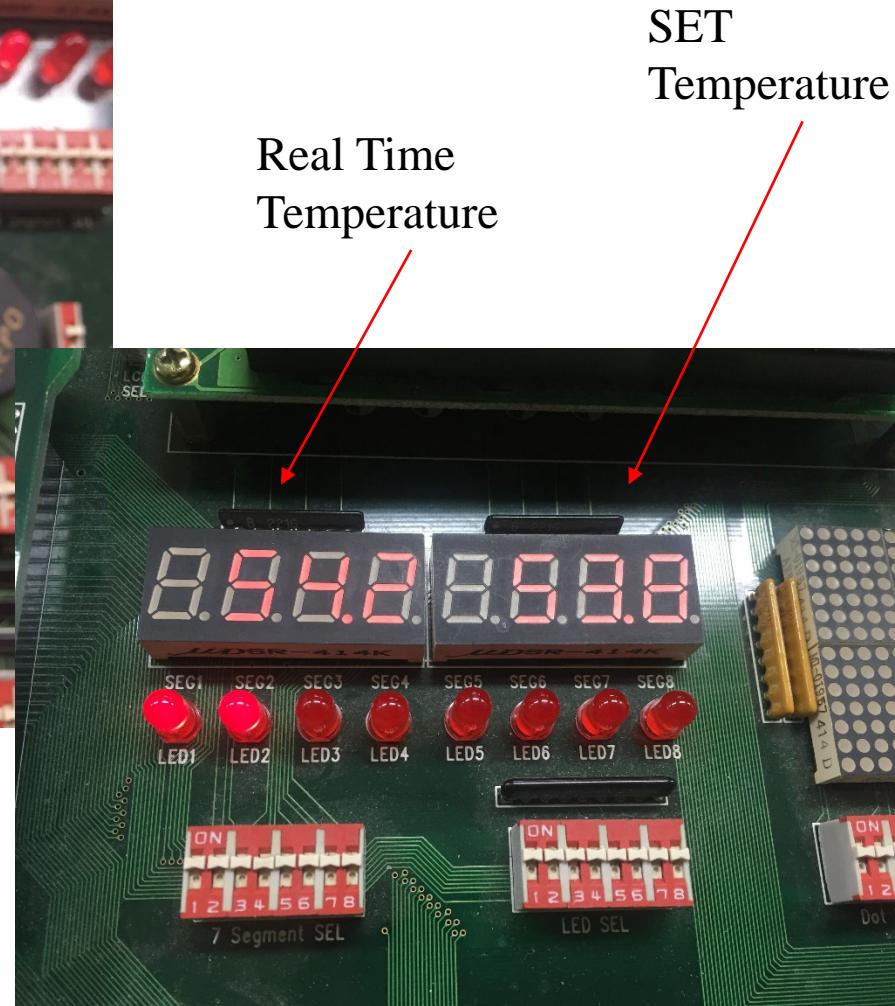
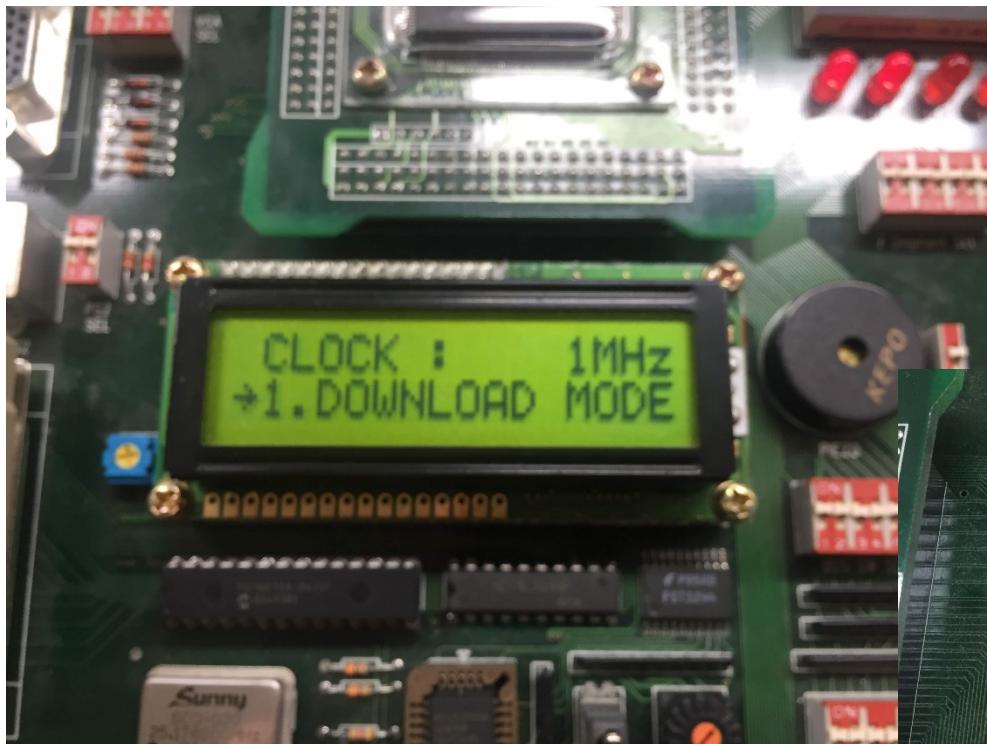


- Changed FPGA Board
- DE2 -> HBE-COMBO 1

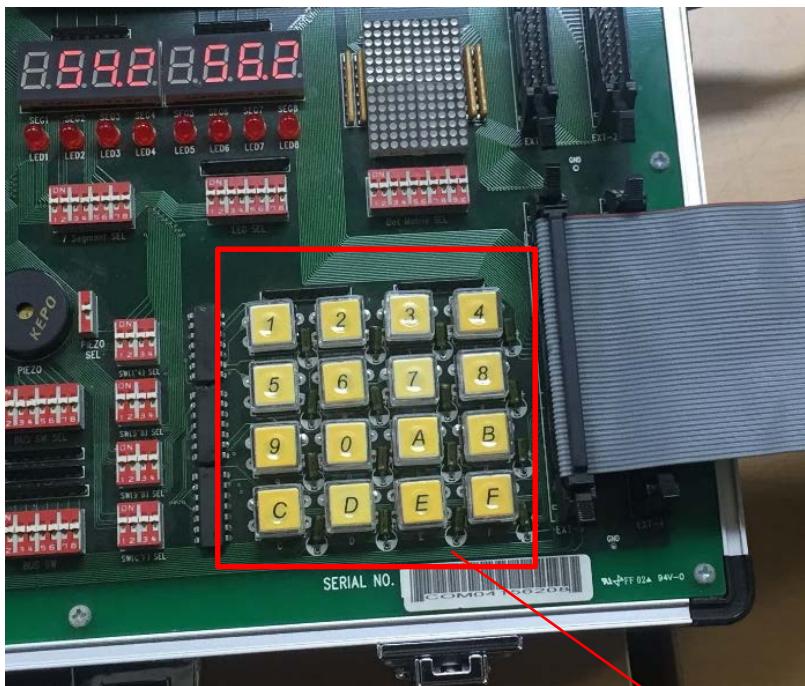


➤ Result

Clock Setting



➤ Result



1	2	3
온도설정	온도상승	온도하강

➤ References

[1] <http://blog.livedoor.jp/hardyboy/tag/FPGA>