

# 결 과 보 고 서

과제명: 오토 캐링 로버(자동 반송 로버)

팀 명: LIGHT ON

목차

1. 과제개요
2. 설계목표
3. 설계 및 제작과정
4. 수정사항

경북대학교 IDEC 2022 창의 회로설계 챌린지

## 1. 과제 개요

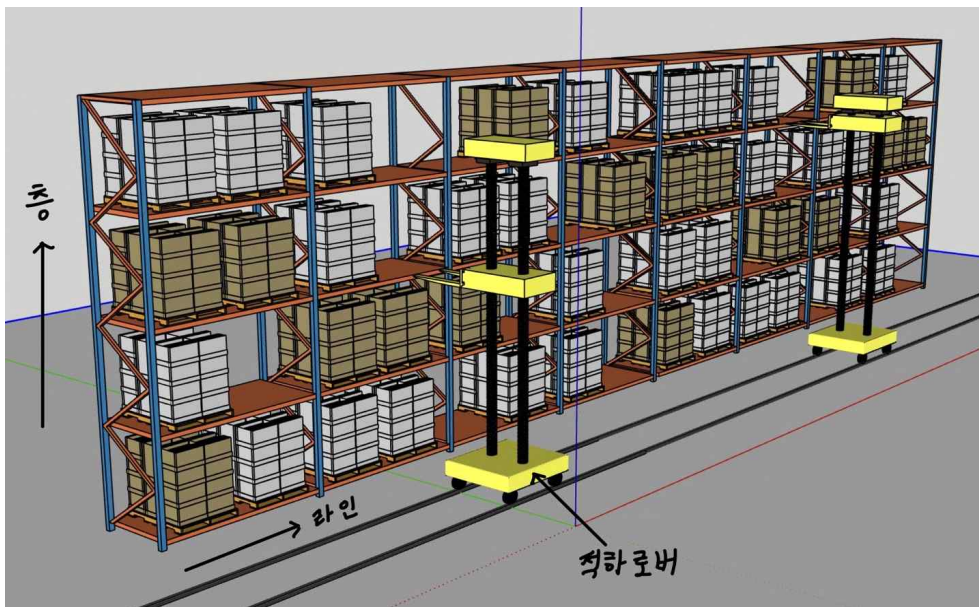
오늘날 수많은 종류의 상품들이 다양한 유형의 물류센터를 통해 끊임없이 입고되거나 출고되고 있다. 또한 COVID-19로 인한 택배 화물 수요가 급증하여 물류 배송에 따른 업무 과중으로 관련 분야 부족 인력 충원의 어려움에 직면해있다. 특히 지역거점 물류센터에서 발생하는 분류, 포장, 재가공, 보관 및 출고 등의 작업들이 많은 인력에 의존하고 있으며 시설노후화는 시급히 개선해야 될 것으로 판단된다. <그림1>은 자동화 물류센터의 적하 시스템을 나타낸 것이다.



<그림1> 자동화 물류센터의 적하 시스템

## 2. 설계 목표

본 과제에서는 소규모 물류센터나 대형상점의 상품 보관 및 배출을 자동화하는 시스템을 micro-controller를 사용하지 않고 Digital logic 및 Analog 소자를 사용하여 구현하고자 한다. <그림 2>는 자동 물품 적하 시스템의 구상도를 나타낸 것이다.



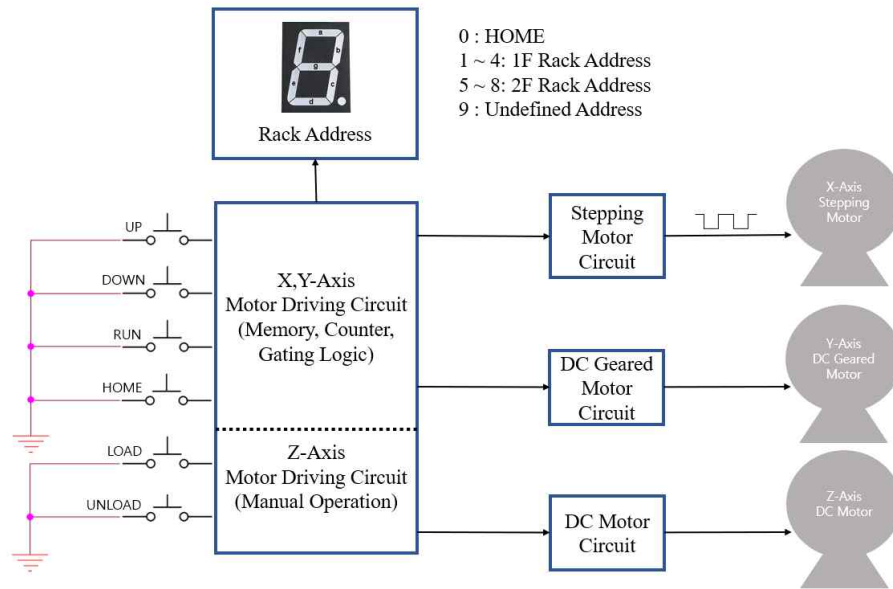
<그림2> 자동 물품 적하 시스템 구상도

### 3. 설계 및 제작과정

#### 1. 시스템 구성도, 동작 알고리즘 설계 및 순서도 작성

##### 1) 시스템 구성도(System Block Diagram)

사용자의 편의와 효율성을 위해 좌표를 입력하면 희망하는 위치로 물품을 저장하거나 하역하기 위해 로버가 이동하도록 하고자 한다. <그림3>은 자동 물품 적하 시스템의 구성도를 나타낸 것이다.



<그림3> 자동 물품 적하 시스템 구성도

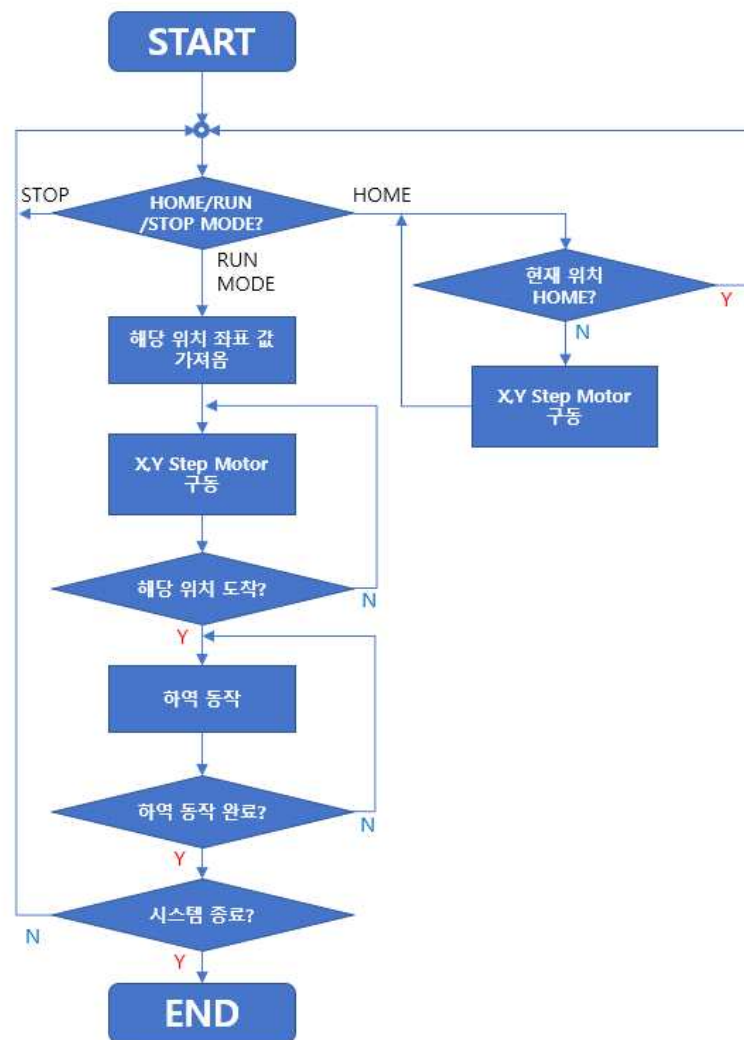
사용자가 UP버튼과 DOWN버튼을 눌러 희망하는 좌표를 입력받아 디코더를 이용하여 7-segment에 표시하고, 입력받은 좌표에 해당되는 공간으로 이동하기 위해 스텝핑 모터가 필요로 하는 회전수를 Memory로부터 입력받도록 회로를 설계한다.

희망하는 위치에 로버가 도착한 후 로버의 적하 ARM에 부착된 모터를 구동시켜 물품을 적재 또는 하역하도록 하며, 동작이 완료되면 초기위치(HOME)로 돌아오도록 한다.

기구부는 X축(Line)과 Y축(Floor) 이동은 레일과 바퀴 및 볼 스크류를 사용하여 구현하고자 한다.

## 2) 순서도

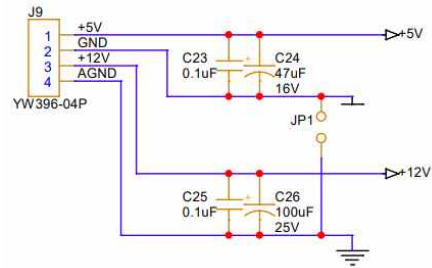
<그림4>는 자동 물품 적하 시스템의 동작 순서도를 나타낸 것이다. 시스템의 초기 설정을 위해 시스템의 초기 설정을 위해 Stop mode에서 HOME 버튼으로 로더를 초기 위치로 가져간다. 이어서 UP/DOWN 버튼으로 원하는 위치를 설정한 후 RUN 스위치를 ON시켜 Run mode로 바꿔 스텝핑 모터를 구동시킨다. 해당 위치에 도착하게 되면 하역 동작을 실행하게 되고 이와 같은 흐름으로 시스템이 반복적으로 동작한다.



<그림4> 자동 물품 적하 시스템의 동작 순서도

## 2. 회로 설계 및 동작원리

### 1) Power Supply(전원부)



전원부에 여러 용량의 커패시터를 병렬로 연결하여 다양한 주파수의 노이즈를 Bypass 시켜준다.

- 전해 커패시터(수 KHz ~ 수 MHz)

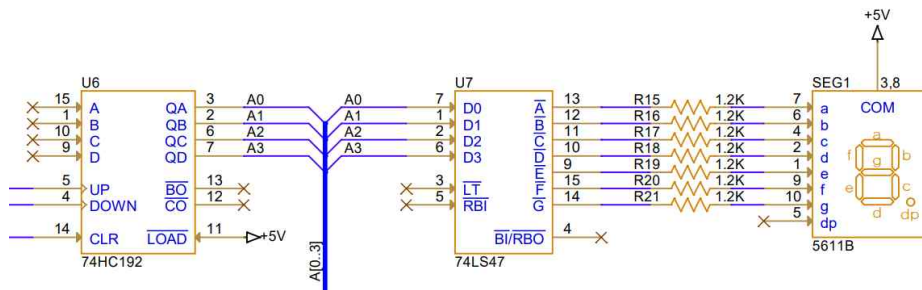
극성이 있으며 커패시터의 용량이 커서 공진주파수가 상대적으로 작아 저주파 노이즈를 잡는데 효율적. 일반적으로 전원부에 자주 사용.

- 모노 세라믹 커패시터(수십 MHz)

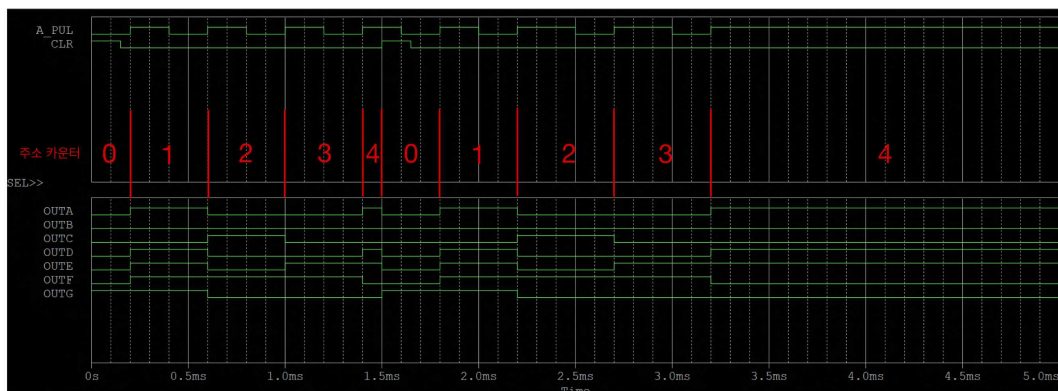
주파수 특성이 좋아 고주파 노이즈를 잡는데 효율적.

일반적으로 IC 신호라인에 자주 사용.

### 2) Address Counter & Decoder and 7-segments(카운터 & 디코더, 7세그먼트)

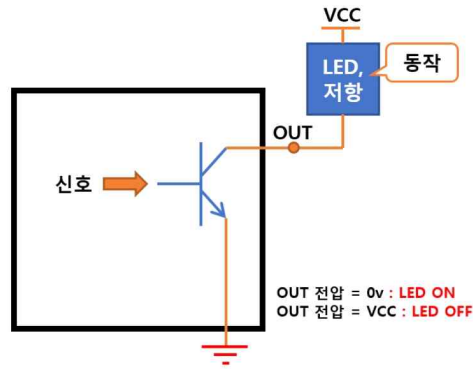


실제 회로에서는 스위치를 사용하여 주소를 변경할 예정이지만 ORCAD 시뮬레이션으로 한눈에 간편히 보기 위해서 카운터(74HC192)를 사용하여 디코더(74LS47)의 출력을 측정해보았다.



A\_PUL 신호에 따라 카운터(74HC192)의 출력이 가르키는 주소가 1씩 증가하면서 디코더(74LS47)를 거쳐 7-Segments로 현재 주소를 출력하는 회로이다.

아래 Plot이 디코더의 출력핀들의 전압을 측정한 것인데 사용하는 디코더가 Open Collector 방식이므로 7-Segments를 Common Anode 형식인 5611B를 사용하기로 했다.

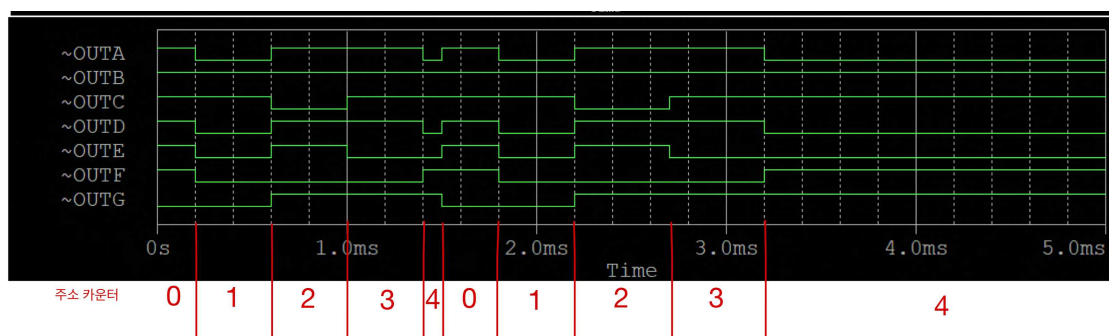


Open Collector 방식인 디코더(74LS47)를 이용해  
세그먼트 내 LED를 끄기 위해서는 Input 신호로 High를 넣어 Transistor를 ON 시켜주면  
Output은 Gnd와 연결되기 때문에 Output은 Low가 출력된다.  
세그먼트 내 LED를 켜기 위해서는 Input 신호로 Low를 넣어 Transistor를 OFF 시켜주면  
Output은 Hi-Impedence 상태가 되고 외부의 풀업 저항 때문에 High로 출력된다.

SN74LS247

LS247  
FUNCTION TABLE

DECIMAL OR FUNCTION	INPUTS						BI/RBO †	OUTPUTS								NOTE
	LT	RBI	D	C	B	A		a	b	c	d	e	f	g		
0	H	H	L	L	L	L	H	ON	ON	ON	ON	ON	ON	OFF	1	
1	H	X	L	LL		H	H	OFF	ON	ON	OFF	OFF	OFF	OFF		
2	H	X	L	LHL			H	ON	ON	OFF	ON	ON	OFF	ON		
3	H	X	L	LH		H	H	ON	ON	ON	ON	OFF	OFF	ON		
4	H	X	L	H	L	L	H	OFF	ON	ON	OFF	OFF	ON	ON		
5	H	X	L	HL		H	H	ON	OFF	ON	ON	OFF	ON	ON		
6	H	X	L	HH		L	H	ON	OFF	ON	ON	ON	ON	ON		
7	H	X	L	HH		H	H	ON	ON	ON	OFF	OFF	OFF	OFF		
8	H	X	H	L	L	L	H	ON	ON	ON	ON	ON	ON	ON		
9	H	X	H	LL		H	H	ON	ON	ON	ON	OFF	ON	ON		
10	H	X	H	LHL			H	OFF	OFF	OFF	ON	ON	OFF	ON		
11	H	X	H	LH		H	H	OFF	OFF	ON	ON	OFF	OFF	ON		
12	H	X	H	H	L	L	H	OFF	ON	OFF	OFF	OFF	ON	ON		
13	H	X	H	HL		H	H	ON	OFF	OFF	ON	OFF	ON	ON		
14	H	X	H	HH		L	H	OFF	OFF	OFF	ON	ON	ON	ON		
15	H	X	H	HH		H	H	OFF	OFF	OFF	OFF	OFF	OFF	OFF		
BI	X	X	X	X	X	X	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	2	
RBI	H	L	L	LL		L	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	3	
LT	L	X	X	X	X	X	H	ON	ON	ON	ON	ON	ON	ON	4	

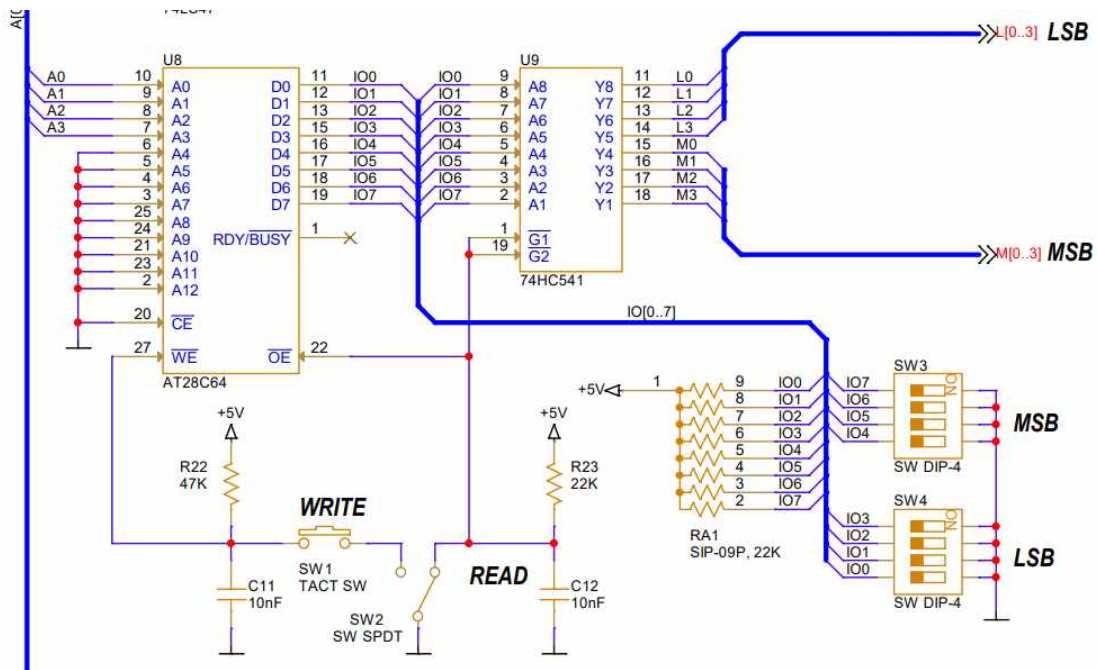


따라서 디코더(74LS47)의 Outputs은 LED의 점등여부와 반대가 되므로 Plot의 Trace에 반전  
기호 “~”를 붙여 반전시켜서 다시 시뮬레이션 해보았다.  
위의 Function Table과 Simulation 결과를 비교해 보면 각 주소가 디코더에 입력되는 시간  
에 알맞게 출력이 나오는 것을 확인해 볼 수 있다.

사진을 보면 0 1 2 3 4 까지 UP COUNT되고 CLR 신호에 의해 초기화 되는 것을 확인할  
수 있는데 이렇게 설정한 이유는 바로 다음에 다룰 Memory(RAM)에서 앞 사이클은 Write로  
뒤 사이클은 Read로 Simulation 할 예정이다.

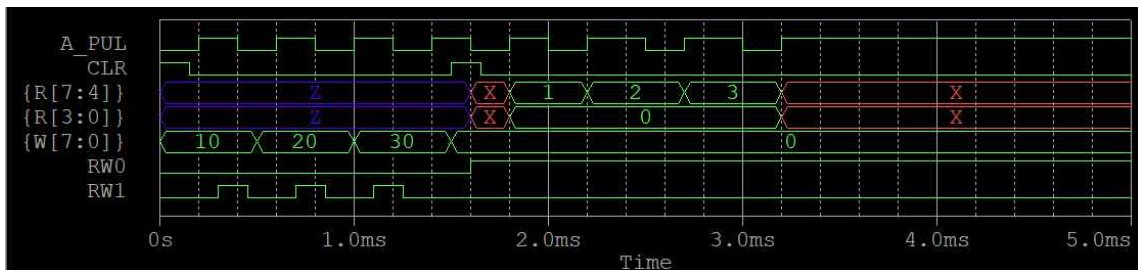


### 3) Memory(메모리)



앞에서 Decoder and 7-segments에서와 마찬가지로 실제 회로에서는 스위치를 사용하여 주소 변경할 예정이지만 ORCAD 시뮬레이션으로 한눈에 간편히 보기 위해서 클럭펄스를 사용하였다. RAM 주소 지정을 위해 카운터(74HC192) 사용하고 RAM 8Kx8b의 출력을 측정해 보았다.

또한 실제회로에서는 EEPROM을 사용할 예정이지만 시뮬레이션을 위해 RAM을 사용하였다.



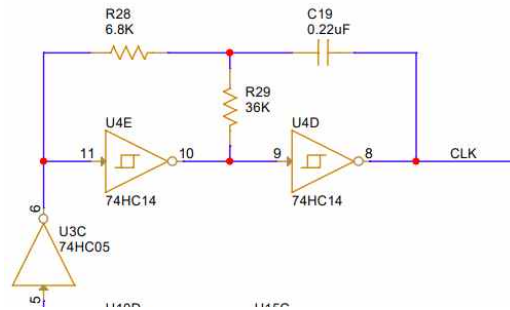
A\_PUL 신호에 따라 카운터(74HC192)의 출력이 가르키는 주소가 1씩 증가하면서 0 1 2 3 4 까지 UP COUNT되고 CLR 신호에 의해 초기화된다. 앞 사이클은 Write로 뒤 사이클은 Read로 Simulation을 하였다.

A\_PUL과 W[7:0]의 타이밍을 고려해 RW1(WE)에 high 신호를 주어 값을 저장하는 방식이며 위의 시뮬레이션은 주소가 0, 1, 2, 3, 4일 때 값 저장X, 10, 20, 30, 값 저장X(8bit이므로 4bit씩 표현, 16진법)을 저장한다.

그 후 CLR신호가 들어온 후 RW1(WE)에 Low 신호를 주고 RW0(RE)에만 High 신호를 주어 현재 주소에 저장된 Data를 읽어온다.

상위 4bit(MSB)는 R[7:4] 으로 출력시키고 하위 4bit(LSB)는 R[3:0]으로 출력시켜 사용할 예정이다.

#### 4) Oscillator(발진기)



프로젝트에서 사용될 스텝모터에 클럭 펄스를 넣어주기 위해 74HC14(Hex Inverters with Schmitt-Trigger Inputs)를 사용하여 발진기(Oscillator)를 사용하였다.

74HC05(Hex Inverters with Open-Drain Outputs)의 INPUT이 HIGH일 때 첫 번째 74HC14의 입력을 Low로 잡아두어 CLOCK을 발생하지 않다가 펄스가 필요할 때만 INPUT에 Low를 넣어주어 OPEN시켜 CLOCK을 발생시킨다.

$$\tau = R_2 \cdot C_1$$

$$V_1 = V_{cc} \cdot (1 - e^{-\frac{t}{\tau}}) - \frac{V_{cc}}{2} \cdot e^{-\frac{t}{\tau}} = V_{cc} - \frac{3 \cdot V_{cc}}{2} \cdot e^{-\frac{t}{\tau}}$$

The switching of the schematic will happen when  $V_1$  becomes equal to  $V_{cc}/2$ , so:

$$\frac{V_{cc}}{2} = V_{cc} - \frac{3 \cdot V_{cc}}{2} \cdot e^{-\frac{t}{\tau}}$$

Or:

$$\frac{V_{cc}}{2} = \frac{3 \cdot V_{cc}}{2} \cdot e^{-\frac{t}{\tau}}$$

$$\frac{1}{3} = e^{-\frac{t}{\tau}} \Rightarrow 3 = e^{\frac{t}{\tau}}$$

$$\ln 3 = \frac{t}{\tau}$$

$$t = \tau \cdot \ln 3 = R_2 \cdot C_1 \cdot \ln 3 = 1.098612289 \cdot R_2 \cdot C_1$$

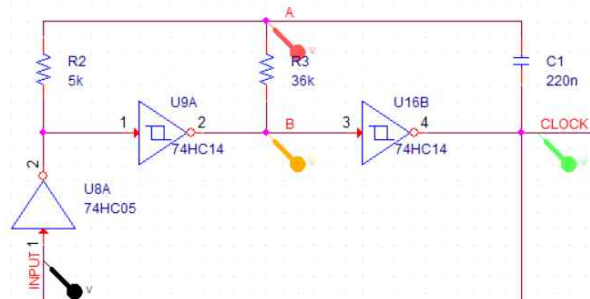
This is the half of the period (because the schematic switches exactly on the half of the  $V_{cc}$ ), so the period:

$$T = 2 \cdot t = 2.197224577 \cdot R_2 \cdot C_1$$

위의 식은 NOT GATE를 활용한 Oscillator의 펄스의 주기를 구하는 식이다. 슈미트 트리거 인버터를 사용한 Oscillator는 약간의 오차가 있을 것으로 예상되지만 위의 식을 사용하여 적당한 값의 저항(36K)과 커패시터(220n)를 선정하였다.

$f = 1/(2 \cdot 2RC)$  주파수 약 57.392 Hz 주기 약 17.424 ms의 펄스를 얻을 것으로 예상된다.

#### OSCILLATOR



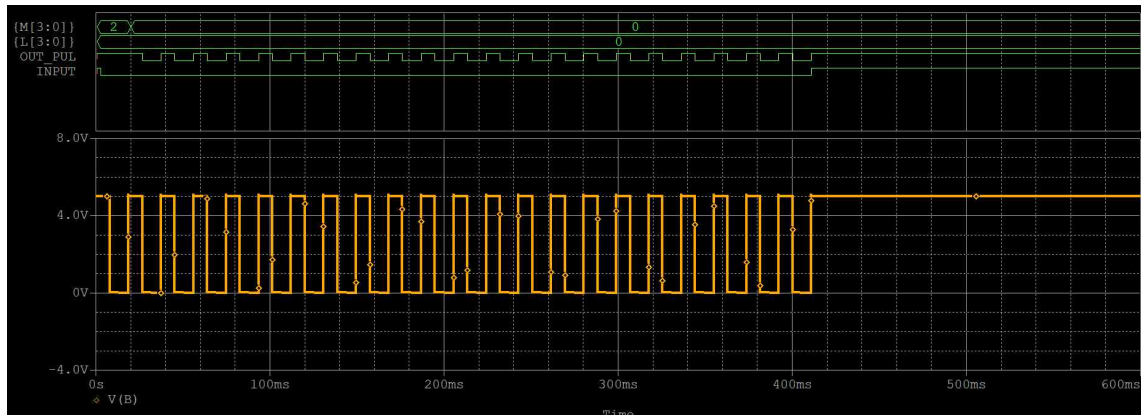




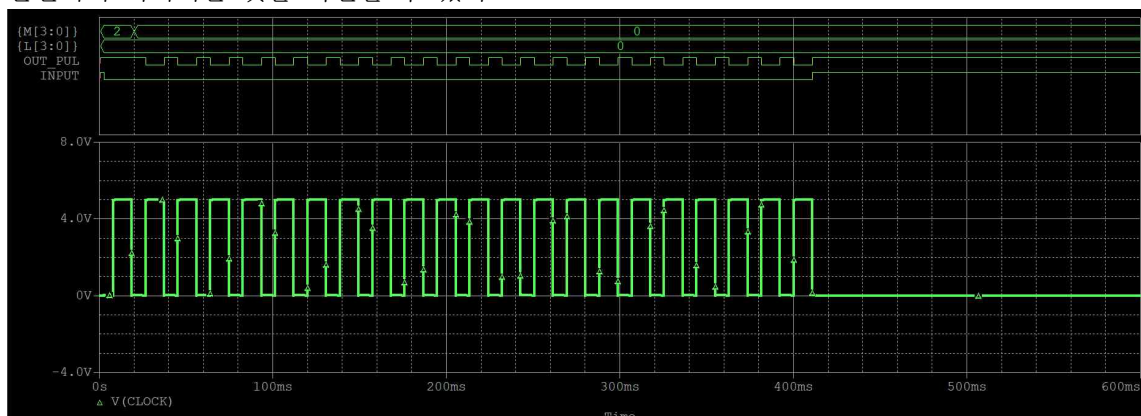
위 파형은 A단의 전압으로 초반 3ms 동안에 74HC05의 INPUT을 High로 잡아두어 첫 번째 74HC14의 입력을 Low로 유지시키는 것을 볼 수 있다. 저항과 커패시터로 인해 충방전 시간만큼 지연되며 그래프를 그리고 주기 약 18 ms의 신호가 나오는 것을 확인할 수 있다.

또한 슈미트 트리거 인버터의 Positive switching threshold(약 2.78)과 Negative switching threshold(약 1.65)를 경계로 값이 반전되면서 전압이 급변하는 것을 확인할 수 있다.

따라서 A단의 두 전압경계를 기준으로 HIGH, LOW가 결정된다.



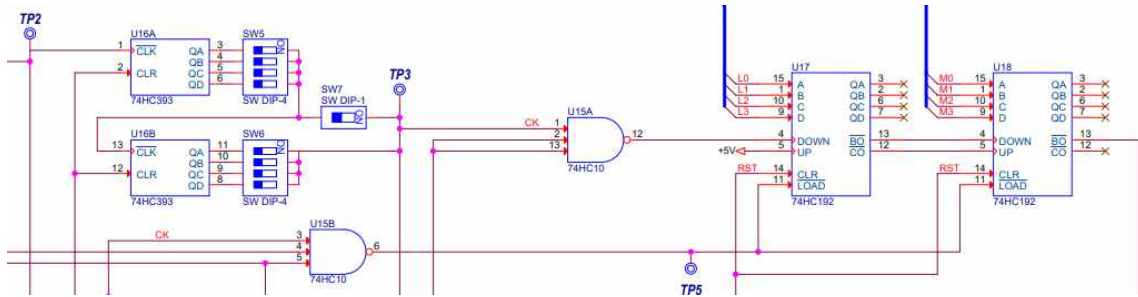
B단의 전압을 살펴보면 위에서 본 A단의 전압이 슈미트 트리거 인버터를 거친 결과이므로 반전되어 나타나는 것을 확인할 수 있다.



다시 한번 B단의 전압이 슈미트 트리거 인버터를 거쳐 우리가 얻고자 하는 CLOCK 펄스를 얻게 되었다. 정리하자면 저항과 커패시터로 충방전 시간만큼 시간을 지연시켜 주기를 늘려 Oscillator를 만들 수 있었다.

최종적으로 출력단(CLOCK)으로 위에서 메모리에 넣어둔 값만큼 DOWN COUNT 하면서 펄스를 발생시키려고 한다.

## 5) X-Axis Stepping Motor Prescaler & Counter(X-축 스텝핑모터 분주기 & 카운터)



REV. A

## HYBRID STEPPING MOTOR TYPE 17HD2603-01N

General specifications		Electrical specifications		Wiring Diagram
Step angle	1.8°±5%	Rated voltage	12 V	
Number of phase	4	Rated current	0.4 A	
Insulation resistance	100Mohm Min.(500V DC)	Resistance per phase	30±10% ohm	
Insulation class	Class B	Inductance per phase	23±20% mH	
Rotor inertia	72 gcm²	Holding torque	220 mNm	
Mass	0.28 kg	Detent torque	20 mNm	

X-축 스텝핑모터로 사용하기로한 17HD2603-01N 의 Step Angle이 1.8도( $0.01\pi$  rad) 이고 사용하는 폴리의 반지름이 8mm이므로

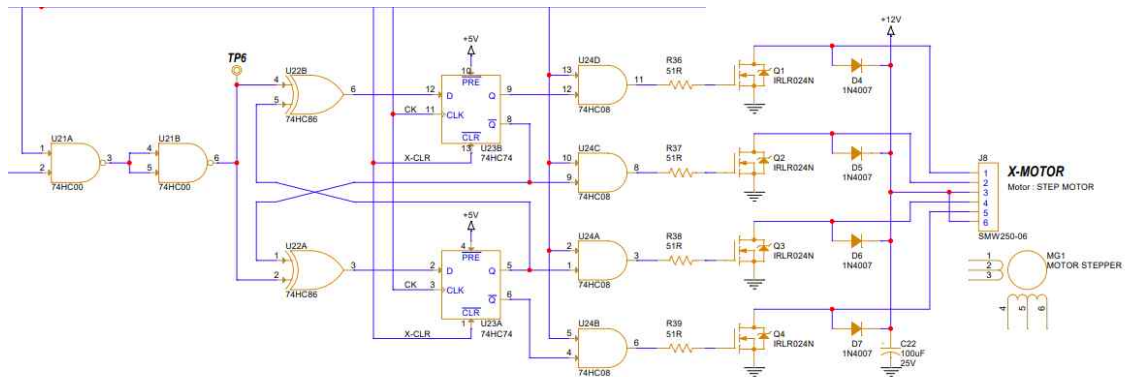
한 펄스 당 움직이는 x-축 이동거리 =  $8 \text{ mm} * 0.01\pi = 0.2513 \text{ mm}$  ( $\because l=r\theta$ )

현재 EEPROM에서 값을 읽어와 MSB와 LSB로 4비트씩 나누어 두자리 수까지 카운트 가능하나 설계한 창고 규격(가로 400mm)에 많이 부족하다.(최대  $0.2513\text{mm} * 99 = \text{약 } 25\text{mm}$ ) 따라서 카운터의 DOWN신호 앞에 분주기(Prescaler)를 달아 최적의 분주비를 찾는 방향으로 설계하였다.

분주비	펄스 당 X-축 이동거리( $l=r\theta$ )	최대이동(400mm)에 필요한 펄스 수
1	$8 \text{ mm} * 0.01\pi = 0.2513 \text{ mm}$	1592
2	$8 \text{ mm} * 0.01\pi * 2 = 0.5026 \text{ mm}$	796
4	$8 \text{ mm} * 0.01\pi * 4 = 1.0052 \text{ mm}$	398
8	$8 \text{ mm} * 0.01\pi * 8 = 2.0104 \text{ mm}$	199
16	$8 \text{ mm} * 0.01\pi * 16 = 4.0208 \text{ mm}$	99
32	$8 \text{ mm} * 0.01\pi * 32 = 8.0416 \text{ mm}$	50
64	$8 \text{ mm} * 0.01\pi * 64 = 16.0832 \text{ mm}$	25
128	$8 \text{ mm} * 0.01\pi * 128 = 32.1664 \text{ mm}$	12
256	$8 \text{ mm} * 0.01\pi * 256 = 64.3328 \text{ mm}$	6

분주비가 너무 크면 X축 이동 제어의 정밀도가 떨어질 수 있고 너무 작으면 펄스수가 많이 필요하므로 현 프로젝트에서는 두 자리 펄스 수로 표시 가능한 16이 적합하다고 판단하였다.

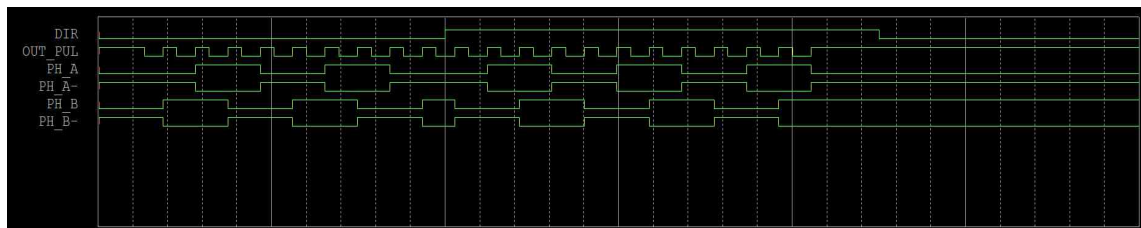
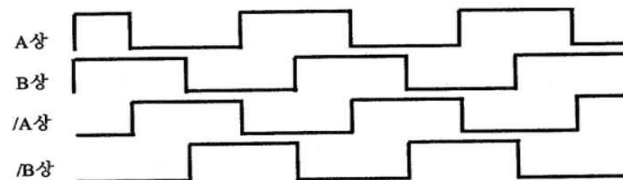
## 6) X-Axis Stepping Motor(모터부)



2상 여자 Full-Step 구동 방식을 사용하여 항상 두 개의 권선이 여자되어 스텝 모터를 구동시키는 방식이다.

XOR게이트와 D-FF를 사용하여 항상 2개의 상이 HIGH이면서 OUT\_PUL이 Positive Edge 일 때마다 순차적으로 아래와 같이 사이클을 가지는 회로를 사용하였다.

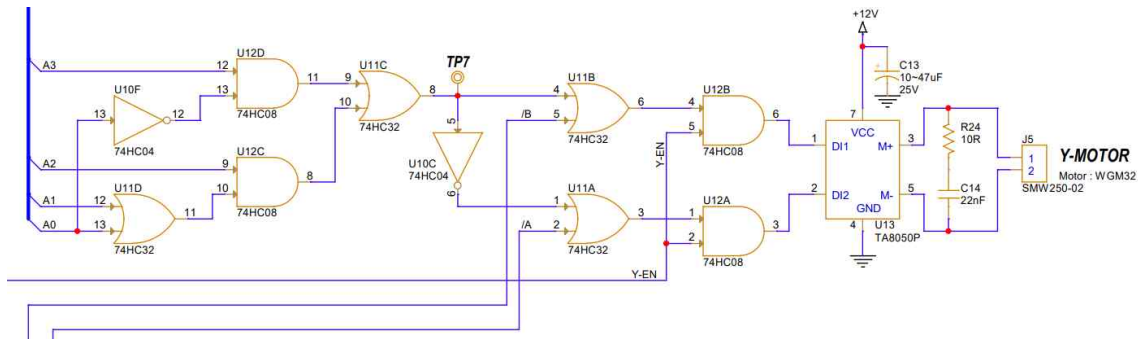
그 후 위에서 다루었던 우측 JK-FF의 Q가 1일 때만 스텝모터를 구동할 수 있도록 AND게이트를 거쳐 권선을 여자한다.



시뮬레이션을 통해 2상 여자 Full-Step 구동 사이클을 확인할 수 있었다.

DIR(회전방향)이 0일 때 반시계방향(CCW) 1일 때 시계방향(CW)으로 회전하는 것을 DIR(회전 방향)이 변경되는 시점을 기준으로 파형이 대칭적으로 존재하는 것을 통해 시각적으로도 쉽게 확인할 수 있었다.

## 7) Y-Axis DC Geared Motor(Y축 DC 기어드 모터)



Y축 이동(1F/2F) 동작을 할 Y-축 모터의 설계 과정은 TP7을 기점으로 전과 후로 나누어 설계하였다. 사용자가 스위치로 설정한 주소 값(A3~A0)에 따라 1F에 해당하는 주소(0, 1, 2, 3, 4)이면 TP7 지점에 0(1F)이 출력되도록 2F에 해당하는 주소(5, 6, 7, 8)이면 1(2F)가 출력되도록 간소화된 논리식을 도출하기 위해 아래와 같이 k-map을 작성하여 설계하였다.

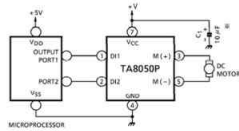
(1) Y(TP7) 논리식 도출

A3 A2 \ A1 A0	00	01	11	10	비고
00	0	0	X	1	A3~A0 : Address 1F : 0 2F : 1
01	0	1	X	0	
11	0	1	X	X	
10	0	1	X	X	

$$\begin{aligned}
 Y(TP7) &= \sum m(5,6,7,8) \\
 &= A2A1 + A2A0 + A3A0' \\
 &= A2(A1 + A0) + A3A0'
 \end{aligned}$$

이제 TP7을 기점 후를 설계하기 위해서 모터드라이브로 선택한 TA8050P의 Datasheet의 Truth Table을 참조하여 TA8050P의 핀1(DI1)과 핀2(DI2)의 출력을 Y(TP7)과 A(2F 리미트스위치), B(1F 리미트스위치)로 간소화된 논리식을 도출하고자 k-map을 활용하였다.

EXAMPLE OF APPLICATION CIRCUIT



※ Connect this capacitor as close to the IC as possible.

PIN DESCRIPTION

PIN No.	SYMBOL	DESCRIPTION
1	DI1	Output status control pin. Connects to a PNP-type voltage comparator.
2	DI2	Connects to the DC motor. Both the sink and the source have a current capacity of 1.5A. Diodes for absorbing counter electromotive force are contained on the VCC and GND sides.
3	M(+)	Connects to the DC motor together with pin 3 and has the same function as pin 3. This pin is controlled by the inputs from pins 1 and 2.
4	GND	Grounded
5	M(-)	Connects to the DC motor together with pin 3 and has the same function as pin 3. This pin is controlled by the inputs from pins 1 and 2.
6	(N.C)	Not connected
7	VCC	Power supply pin. This pin has a function to turn off the output when the applied voltage exceeds 27.5V, thus protecting the IC and the load.

TRUTH TABLE

Input		Output	
DI1	DI2	M (+)	M (-)
H	H	L	L
L	H	L	H
H	L	H	L
L	L	OFF (high impedance)	

Note : Brake mode comes into effect when both M(+) and M(-) go low ; stop mode comes into effect when both M(+) and M(-) turn OFF.

<TA8050P Datasheet>

INPUT		OUTPUT		
DI1	DI2	M(+)	M(-)	DIR
H	L	H	L	CW
L	H	L	H	CCW
L	L	OFF		

우리가 사용하고자 하는 입출력 진리표를 다시 정리하여 배경색별로 모드를 k-map에 표시하였다.

1F -> 2F : CW(DI1(H) DI2(L))

2F -> 1F : CCW(DI1(L) DI2(H))

STOP : DI1(H) DI2(H)

(2) TP7 기점 후 DI1 논리식 도출

A B Y(TP7)	00	01	11	10	비고
0	1	0	0	1	A : 2F 도착시 0 B : 1F 도착시 0 Y : 설정주소가 1F이면 0 2F이면 1
1	1	1	1	1	

$$DI1 = \sum m(0,1,3,4,5,7) \\ = Y + B'$$

1F -> 2F : CW(DI1(H) DI2(L))

2F -> 1F : CCW(DI1(L) DI2(H))

STOP : DI1(H) DI2(H)

(3) TP7 기점 후 DI2 논리식 도출

A B Y(TP7)	00	01	11	10	비고
0	1	1	1	1	A : 2F 도착시 0 B : 1F 도착시 0 Y : 설정주소가 1F이면 0 2F이면 1
1	1	1	0	0	

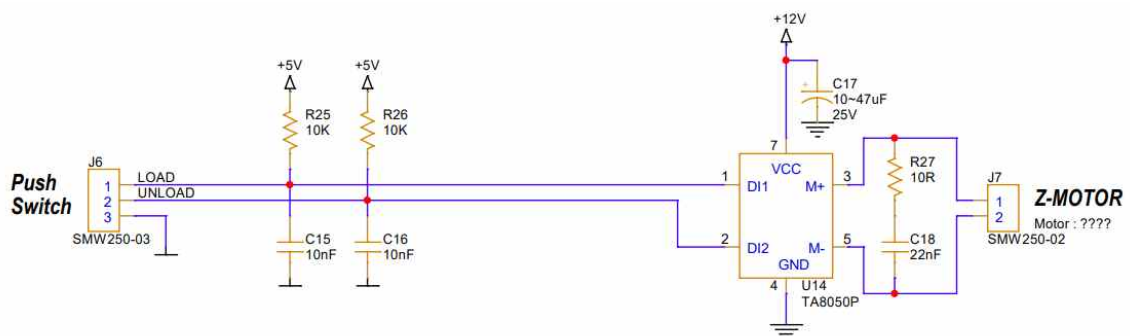
$$DI2 = \sum m(0,1,2,3,4,6) \\ = Y' + A'$$

1F -> 2F : CW(DI1(H) DI2(L))

2F -> 1F : CCW(DI1(L) DI2(H))

STOP : DI1(H) DI2(H)

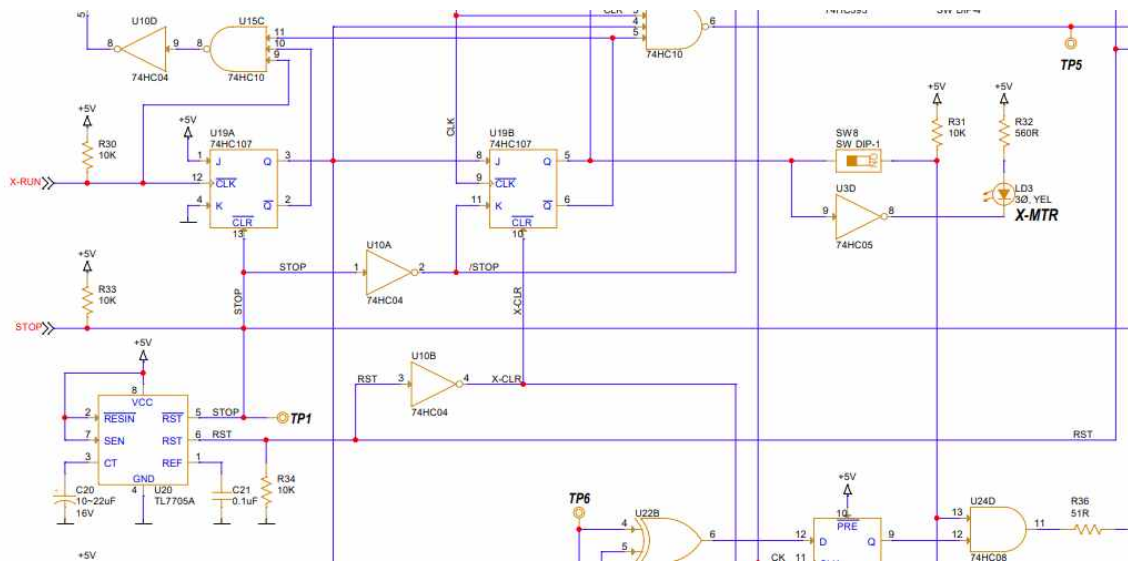
8) Z-Axis DC Geared Motor(Z축 DC 모터)



적재/하역 동작을 할 Z-축 모터는 로버가 설정한 위치에 도착하면 택트 스위치를 사용하여 수동 조작하는 방식으로 사용할 계획이다. Y-축과 마찬가지로 모터드라이브(TA8050P)를 사용하여 모터의 동작과 회전 방향을 제어하고자한다. LOAD 스위치를 누르면 모터드라이브의 입력이 DI1(L) DI(H)가 되어 CCW으로 모터가 회전하여 리드스크류를 따라 적재 동작을, UNLOAD 스위치를 누르면 모터드라이브의 입력이 DI1(H) DI(L)가 되어 CW으로 모터가 회전하여 리드스크류를 따라 하역 동작을 수행하는 개념이다.



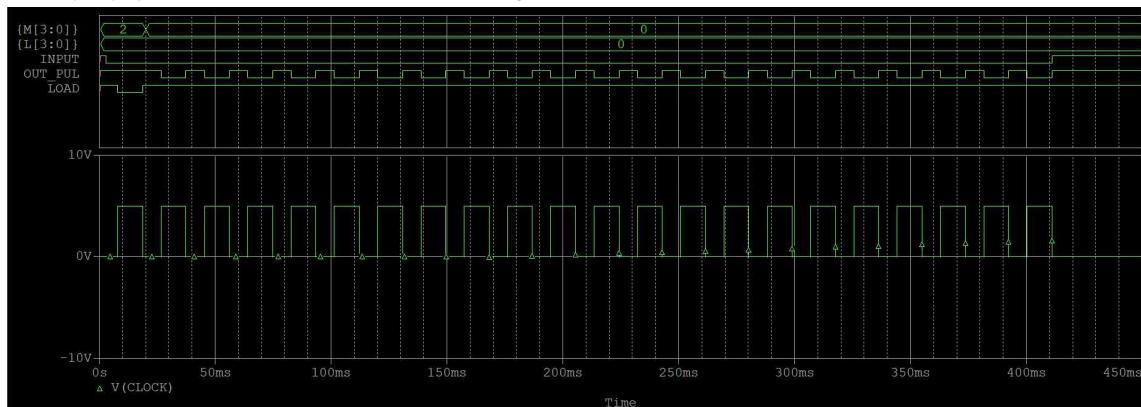
## 9) Gating Logic



JK-FF과 NAND gate를 활용하여

(왼쪽 JK-FF) : 시스템 초기와 MSB Counter의 값을 모두 Down Count 하여 MSB(우측 카운터)의 BO가 LOW로 되는 시간에 JK-FF를 Clear 시킴으로써 Oscillator를 동작을 중지시키기 위해서 U15C(NAND Gate)가 펄스를 출력 하지 못하도록 제어한다.

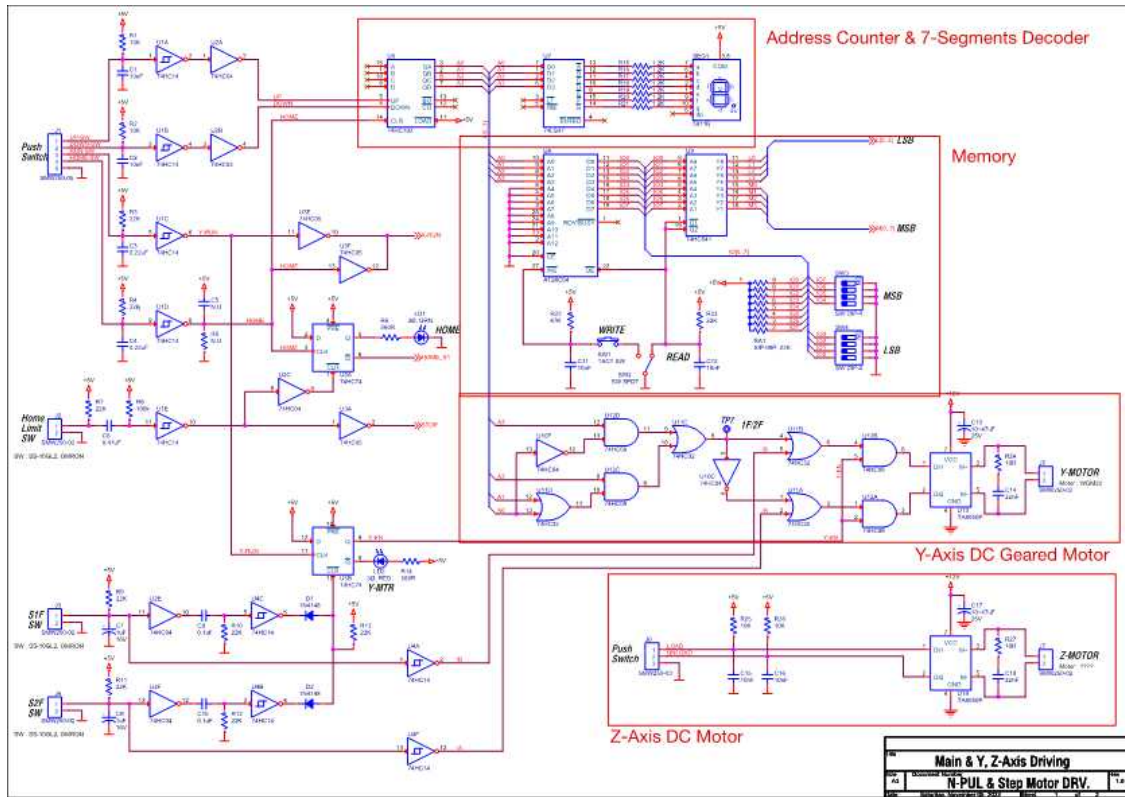
(오른쪽 JK-FF) : CLOCK이 처음 High가 되고 첫 Negative Edge Trigger가 발생하는 시점까지 EEPROM에 저장된 값만큼 펄스를 생성하기 위해 READ한 EEPROM의 값(MSB/LSB)을 4bit씩 각각 카운터에 LOAD하기 위해 사용한다.



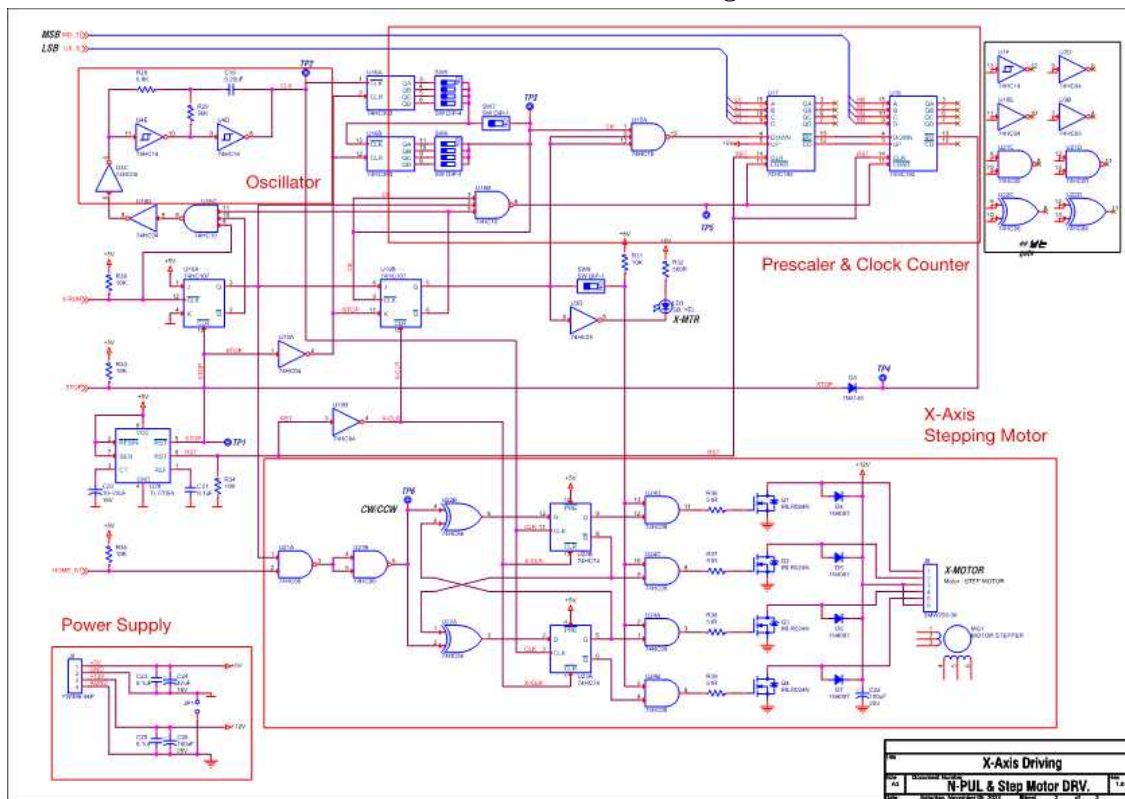
시스템 초기와 20개의 OUT\_PUL의 펄스가 끝난 후 HIGH가 되므로 필요할 때만 Oscillator를 사용할 수 있다. 첫 CLOCK 펄스동안 EEPROM에 저장된 값을 카운터에 LOAD하는 것을 알 수 있다. (Active Low)



9) 전체 회로



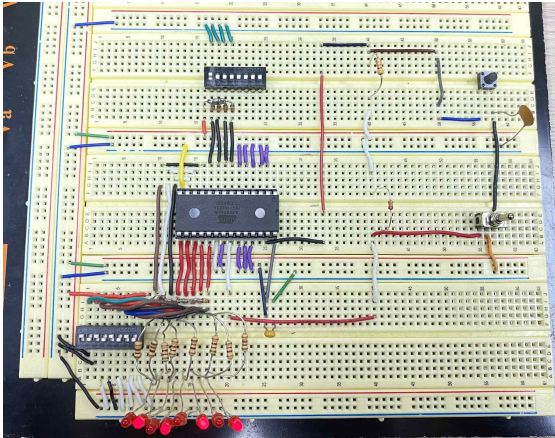
### Main & Y,Z-Axis Driving



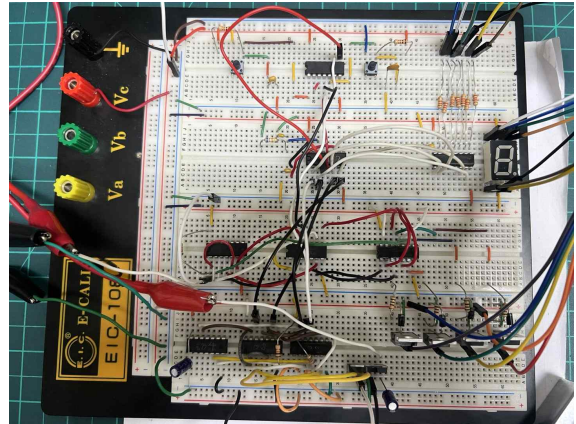
### X-Axis Driving

### 3. PCB ARTWORK 및 납땜 작업

#### 1) 브레드보드 구현

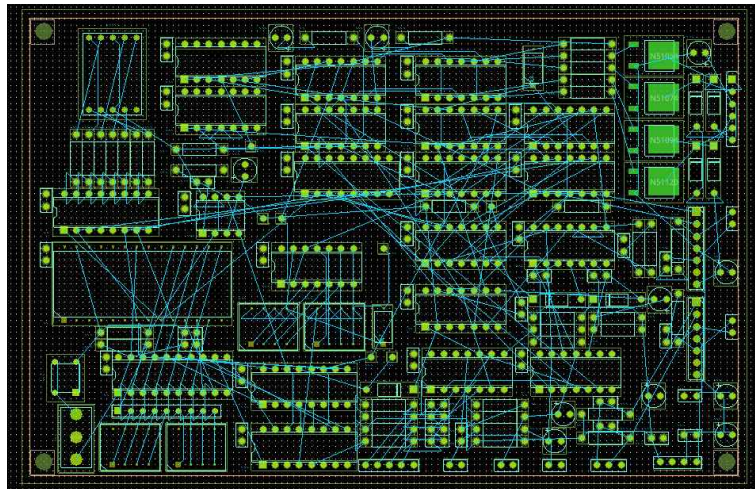


<EEPROM RD/WR Circuit>

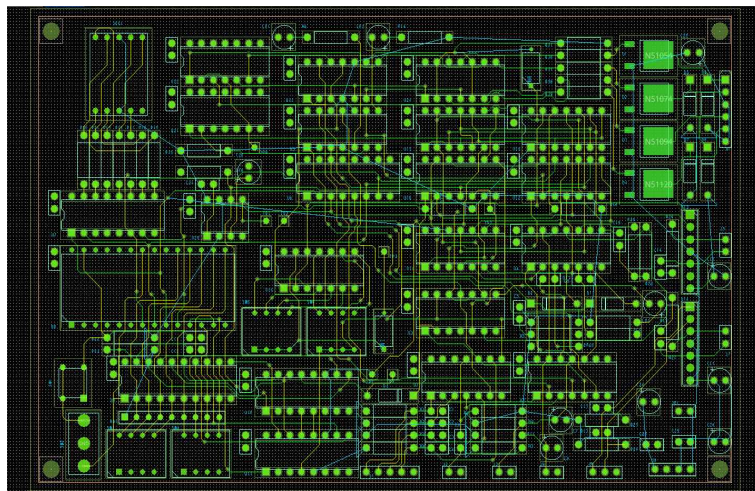


<X-Axis Motor Driving Circuit>

#### 2) PCB

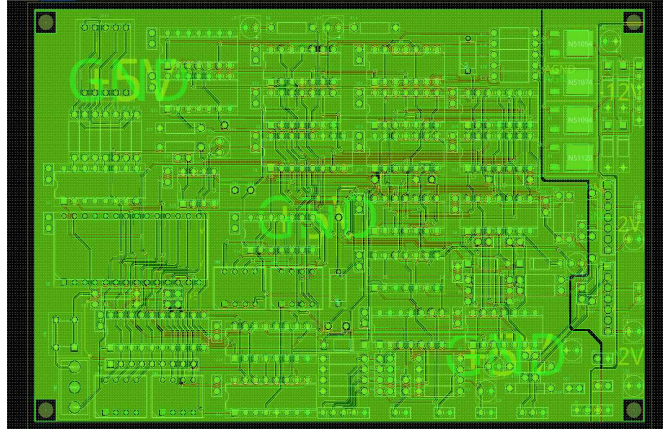


<부품 배치(Placement)>



<배선작업(Routing)>

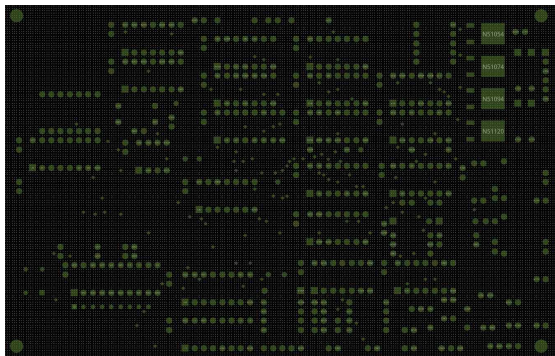




<동판 배치(Cooper)>

DRILL CHART: TOP & BOTTOM			
ALL UNITS ARE IN MILLIMETERS			
FIGURE	FINISHED SIZE	PLATED	QTY
	0.3302	PLATED	124
A	0.7	PLATED	2
A	0.7	PLATED	2
A	0.750	PLATED	1
A	0.750	PLATED	27
A	0.76	PLATED	28
A	0.76	PLATED	4
A	0.8636	PLATED	9
A	0.8649	PLATED	10
A	0.9144	PLATED	368
A	0.9652	PLATED	7
A	1.0	PLATED	3
A	1.0	PLATED	1
A	1.0668	PLATED	158
A	1.2601	PLATED	12
A	1.2601	PLATED	2
A	1.85	PLATED	3
G	3.175	NON PLATED	4

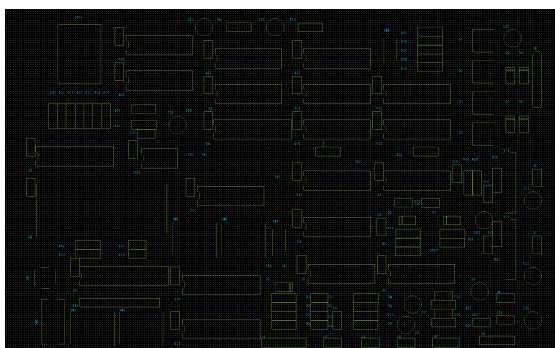
<거버 파일(Gerber File)>



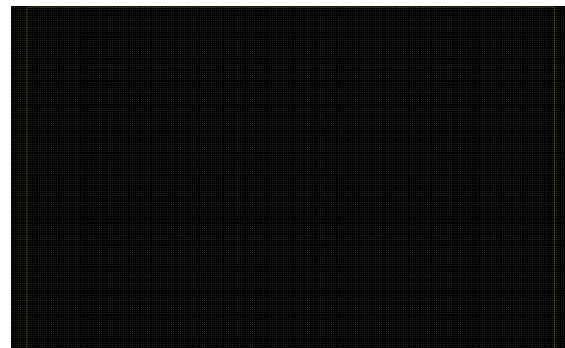
<Soldermask\_TOP>



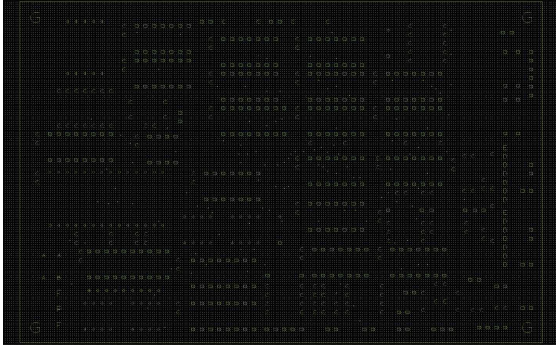
<Soldermask\_BOTTOM>



<Silkscreen\_TOP>

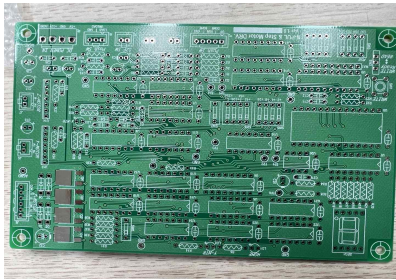


<Outline>

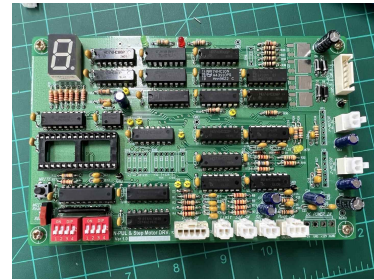


<Drill>

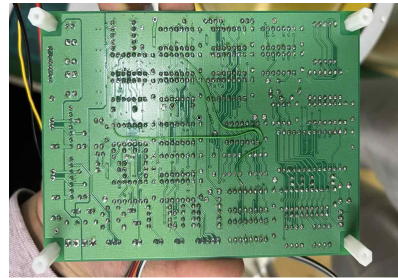
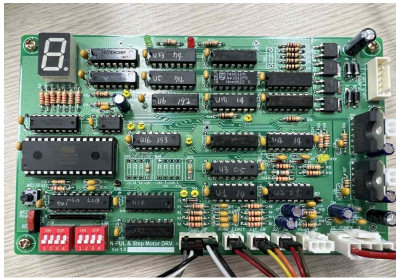
### 3) 납땜 작업



<PCB>



<납땜 작업 중>

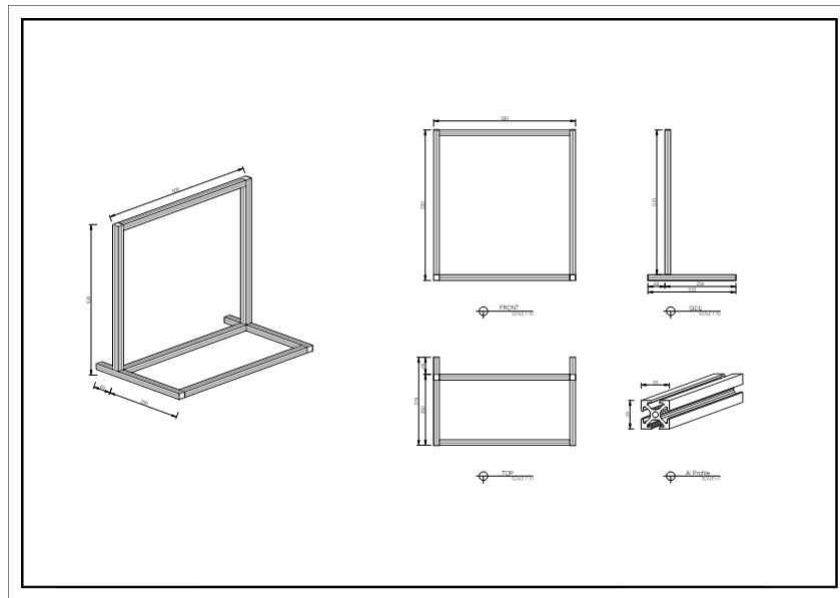


<납땜 최종 완료 및 수정>

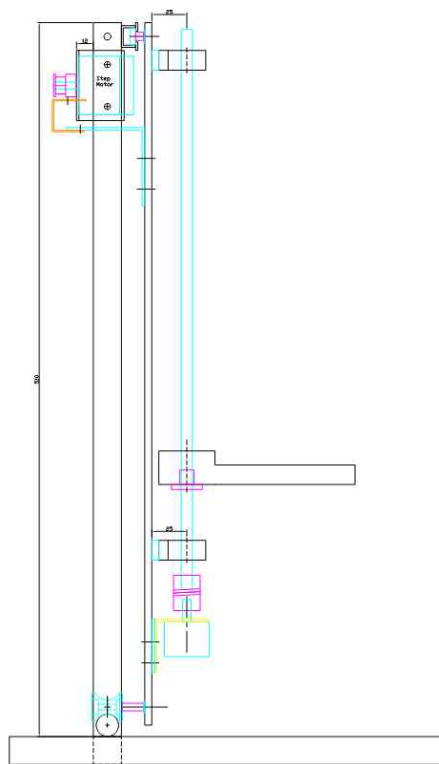


#### 4. 기구부

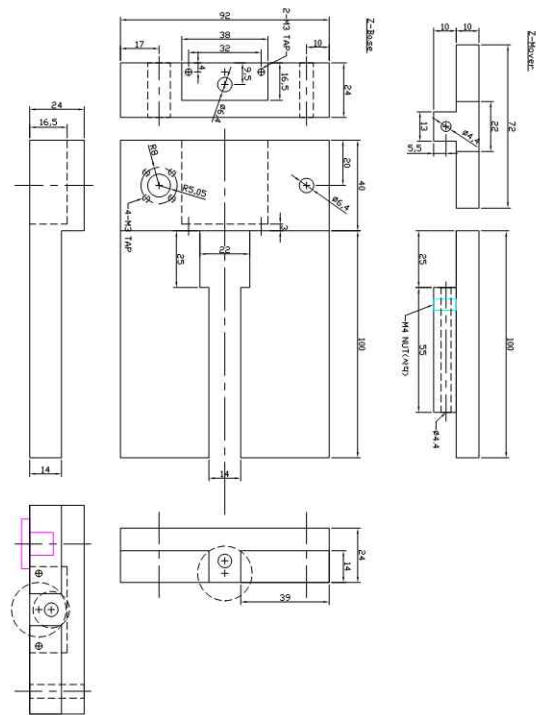
##### 1) 기구부 설계



<AL-프로파일 구조물 도면>



<로버 측면 도면>



<Z-축 로버 도면>

## 2) 기구부 재료 구비



AL-프로파일 구조물



3D 프린터용  
레일 세트

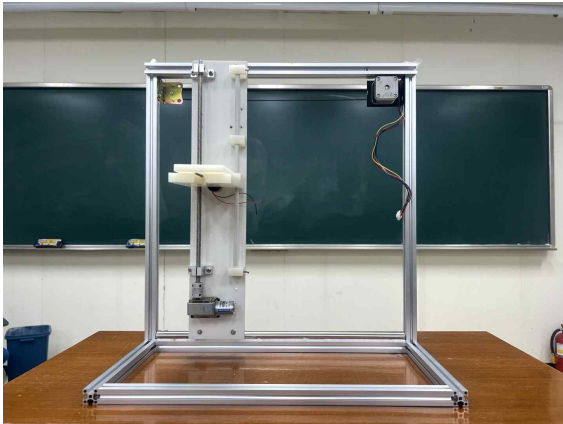


리드 스크류,  
포맥스

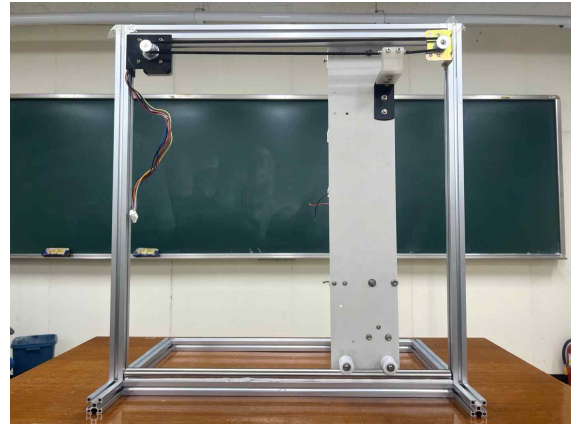


6 to 8 커플링

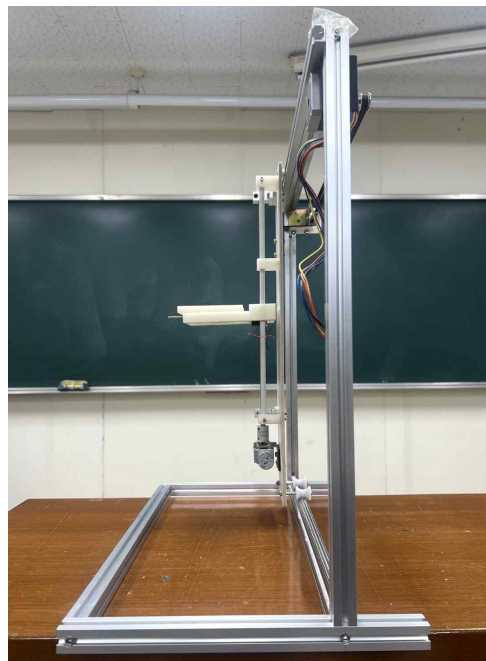
## 3) 기구부 조합 및 제작



<전면>



<후면>



<측면>



#### 4. 수정사항

- (1) Clear overrides load, data and count inputs.
- (2) When counting up the count down clock input (CP<sub>D</sub>) must be HIGH, when counting down the count up clock input (CP<sub>U</sub>) must be HIGH.

##### Sequence

Clear (reset outputs to zero);  
load (preset) to BCD seven;  
count up to eight, nine,  
terminal count up, zero,  
one and two;  
count down to one, zero,  
terminal count down, nine,  
eight, and seven.

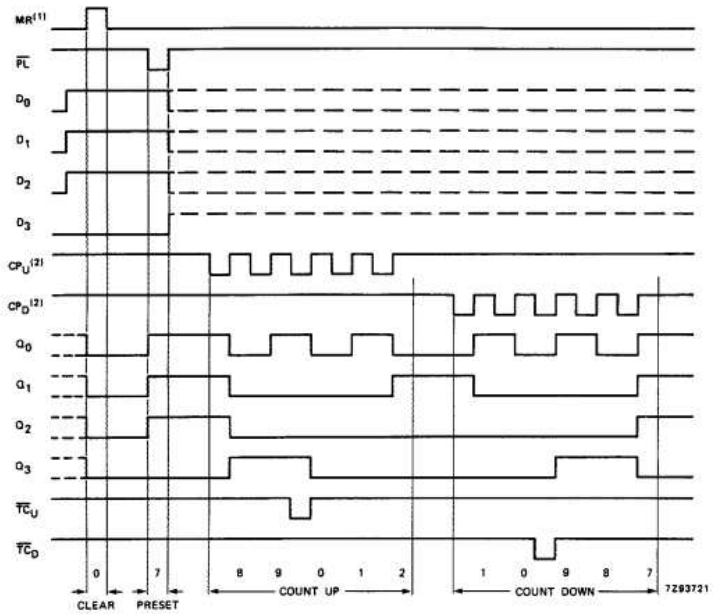
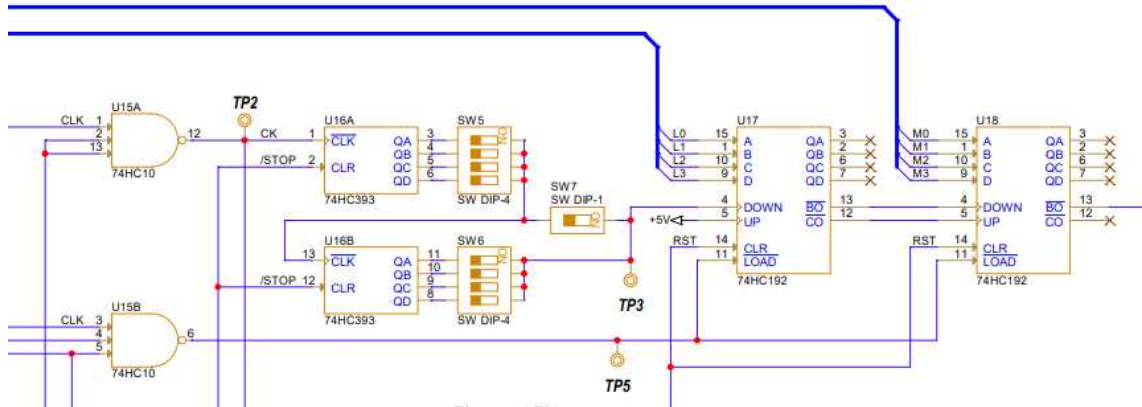


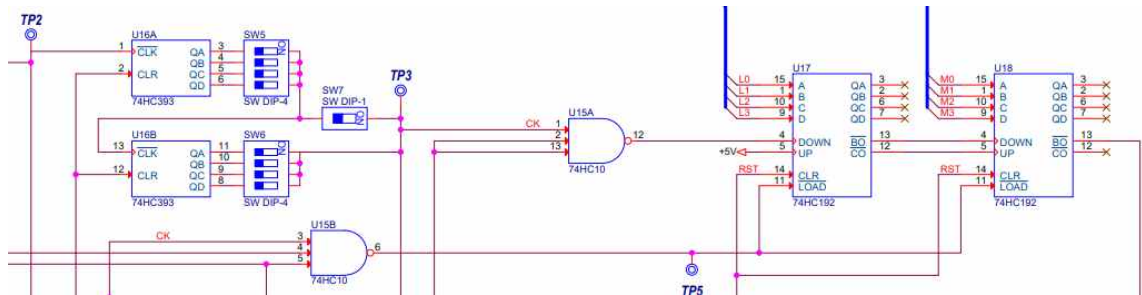
Fig.5 Typical clear, load and count sequence.

##### <74HC192 DataSheet>

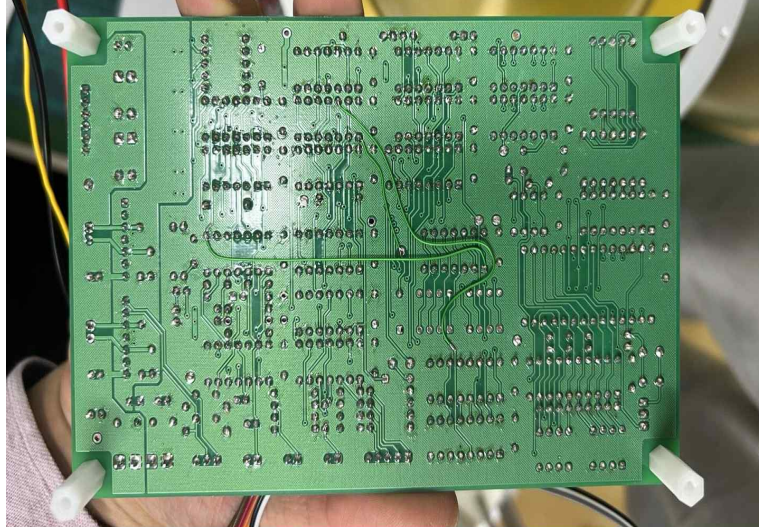
74HC192(10진 카운터)의 DataSheet를 보면 Count DOWN / UP 사용하려면 평상시 HIGH 상태를 유지해야 한다고 하는데 수정 전 회로에서는 평상시 LOW 상태로 설계하였다. 따라서 클럭 신호를 차단하는 U15A(74HC10, NAND)을 분주기 부분 다음으로 옮겨 해당 문제를 해결하였다.



<수정 전>

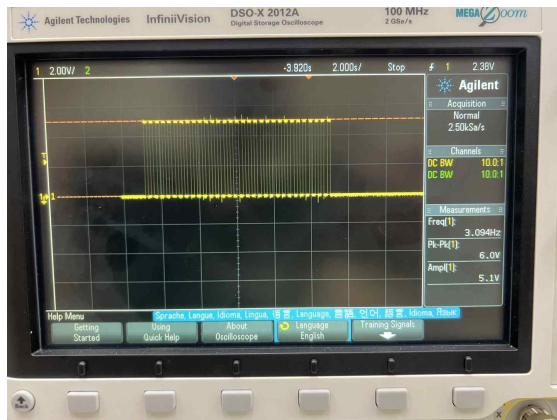


<수정 후>

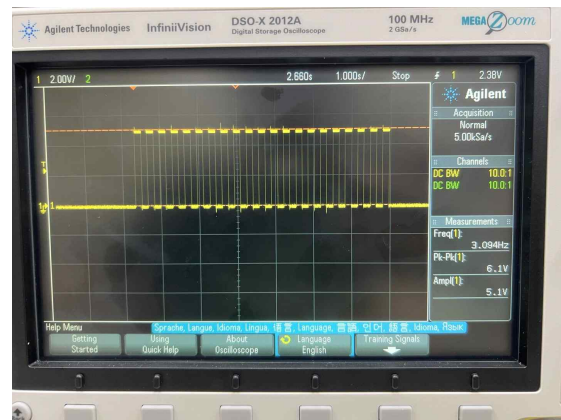


<점프선을 사용하여 회로 수정>

이후 정상 출력되는지 오실로스코프를 이용해 TP3의 파형을 측정하여 확인하였다.



<메모리 10 저장 후 TP3 출력파형>



<메모리 20 저장 후 TP3 출력파형>