

Database Term Project	
이름	2017170989 주동현

1. Project Outline

1.1. Outline

주어진 beauty salon code 에 대해

- 1) Schema 재설정을 통한 기능 보완
- 2) 미용실 예약, 고객용/날짜 별 예약 조회 기능 완성
- 3) 손님 예약 취소 기능 추가

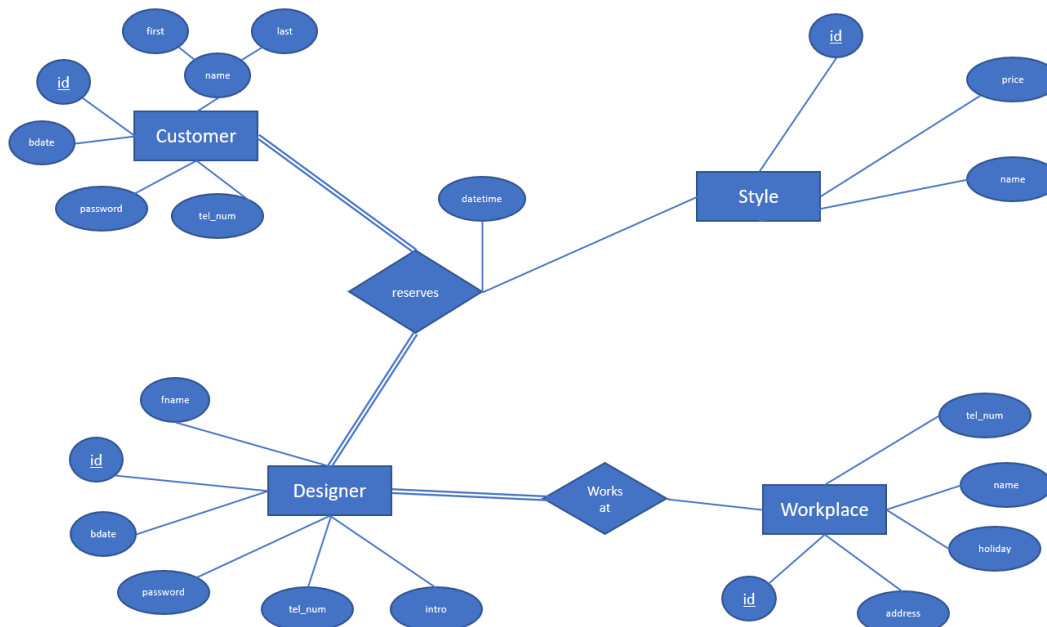
를 수행하였다.

1.2. Project Setup

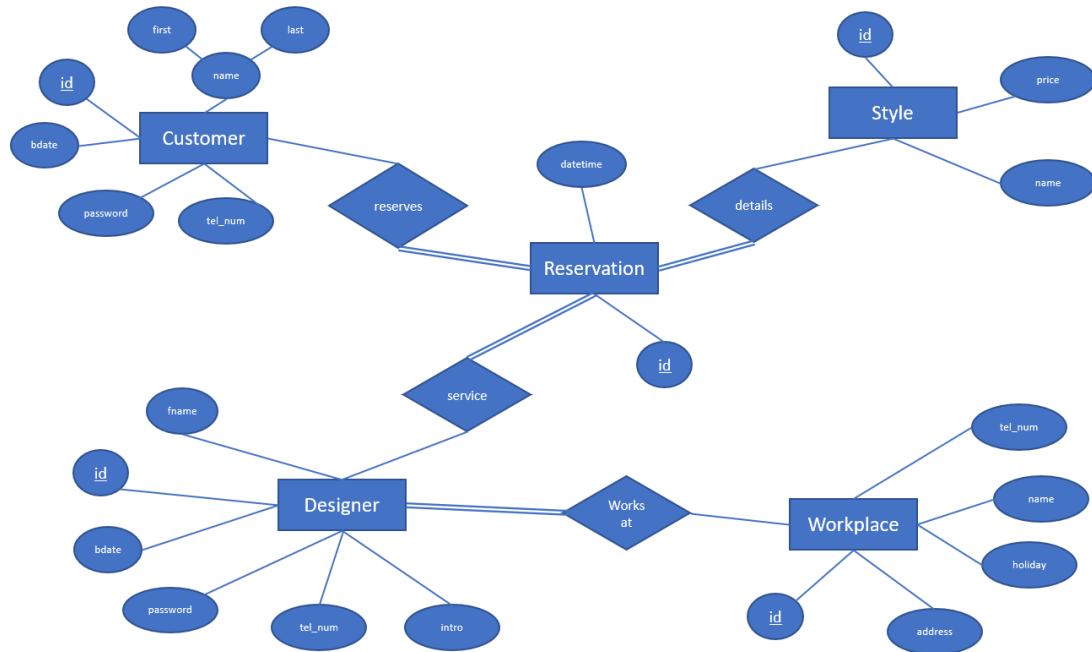
Autoset 프로그램을 통해 웹 서버와 MariaDB 를 모두 구축한다. MariaDB 를 통해 DB engine 관리, html-php 코드를 통해 웹페이지를 관리한다. Html 은 웹사이트 중 정적인 부분을 담당, php 는 웹사이트 중 동적인 부분을 담당하며, user interaction 수행, user input 을 DB query 로 만들어 DB 와 소통하는 역할을 수행한다.

1.3. ER Diagram

Beauty salon DB 를 위해 다음과 같은 ER diagram 을 만들었다.



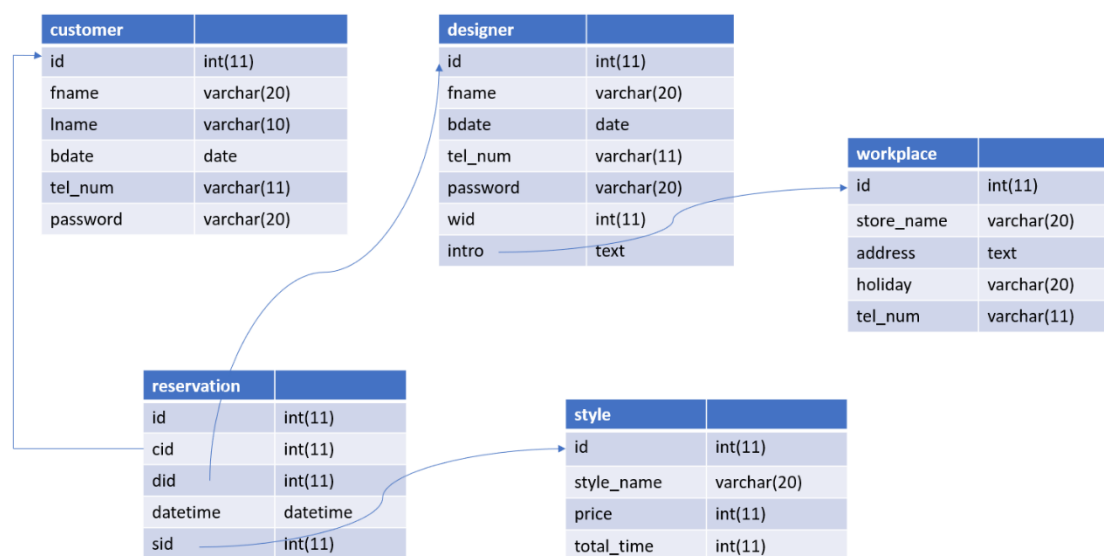
이 때 'Reserve' 관계에는 customer 와 designer, style 이 모두 개입하는 ternary relationship 이다. 이는 entity 사이의 관계를 명확하게 표현하지 못하기에, 'Reserve'를 'Reservation' entity 로 변환하여, ternary relationship 를 3 개의 binary relationship 으로 변환하였다.



기존 schema 에 비교하여 다음 entity, attribute 를 추가하였다.

- 1) Customer, designer entity 의 login 을 위한 password attribute
- 2) Customer entity 의 first name, last name attribute 의 분리
 - a. Designer 의 경우 실생활에서도 first name 만으로 지칭하는 경우가 많아 first name 만 등록한다.
- 3) Price entity 를 style identity 로 변경, designer entity 와의 dependency 를 제거
 - a. 실생활의 미용실을 반영했을 때, designer 별 style 이 정해져 있기보다 미용실 전체에 적용되는 style 이 적절하다고 판단하였다.
- 4) 미용실 지점 정보를 나타내는 workplace entity 의 추가

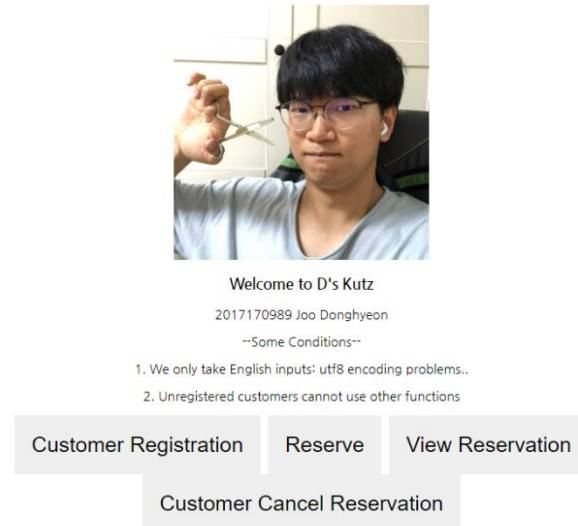
이에 따른 schema 는 다음과 같다.



2. Project Menus and Source Codes (Web-side)

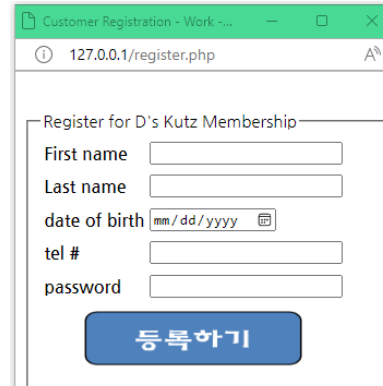
2.1. Main Page (main.php)

메인 페이지는 다음과 같다.



각 버튼은 onClick 속성을 통해 function call을 수행하며, 각 function은 그에 해당하는 .php, 즉 팝업 페이지를 연다.

2.2. Customer Registration (Register.php, Register2.php)



회원 authentication 을 '이름-전화번호' 대신 '전화번호-비밀번호'으로 진행하여 보안을 높이고 개인정보를 보호한다. 이를 위해 회원 등록 시 이름과 생년월일, 전화번호와 함께 비밀번호를 등록한다.

Main.php 에서 버튼 클릭으로 호출되는 register.php 는 위 팝업을 통해 각 정보를 입력 받는다. 이후 '등록하기' button click 을 통해 form 을 register2.php 로 제출한다.

Register2.php 에서 우선 앞에서 입력한 정보를 모두 변수로 받아온다. 입력되지 않은 변수에 대해서 echo 를 통해 재입력을 촉구한다. ("history.go(-1)")

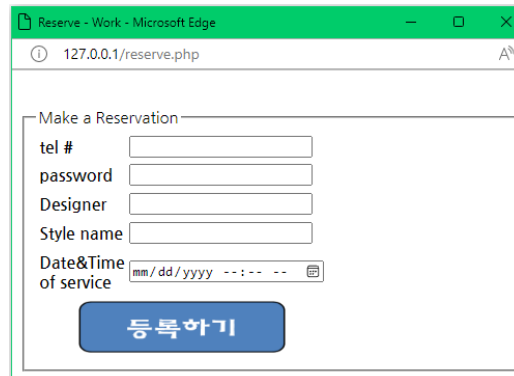
적절한 입력에 대해서 우선 query 를 통해 이미 등록되어 있는 회원인지 확인한다.

```
$sql1 = "select * from customer where lname='$lname' and fname='$fname' and telephone_number='$telephone_number'";
```

이후 insert 를 통해 customer table 에 새로운 row 를 입력한다.

```
$sql2 = "insert into customer(fname, lname, bdate, tel_num, password) values('$fname', '$lname', '$date', '$telephone_number', '$password')";
```

2.3. Make Reservation (Reserve.php, Reserve2.php)



회원 본인만이 예약을 할 수 있도록, 또한 회원 간의 구별을 위해 "전화번호-비밀번호"를 통해 회원을 식별한다. 디자이너 이름과 스타일, 예약 날짜-시간을 통해 예약을 진행한다.

Reserve.php 는 register.php 과 동일한 방식으로 입력을 받아 form 을 reserve2.php 로 제출한다.

Reserve2.php 도 우선 register2.php 와 동일한 방식으로 모든 변수가 입력되었는지 확인한다. 이후 customer table 로의 query 를 통해 알맞은 회원 정보를 불러온다.

```
$sql1 = "select * from customer where password='$password' and tel_num='$telephone_number'";
```

일치하지 않는 회원 정보의 경우 그에 맞는 알림을 띄운다.

```
$result1 = mysqli_query($connect,$sql1);
$num1 = mysqli_num_rows($result1);

if($num1==0){
    echo "<script>window.alert('누구세요?'); history.go(-1);</script>";
    exit;
}
```

Sql1 query 를 통해 회원 식별자인 cid 를 불러온다.

```
$cid = mysqli_fetch_row($result1)[0];
```

동일한 방식으로 디자이너 식별자 did 를 포함한 디자이너 정보, 스타일 식별자 sid 를 포함한 스타일 정보를 불러온다. (스타일 코드 사진 생략)

```
//check if existing designer
$sql_designer = "select * from designer where fname='$designer_name'";
$result_designer = mysqli_query($connect,$sql_designer);
$num_designer = mysqli_num_rows($result_designer);

if($num_designer==0){
    echo "<script>window.alert('존재하지 않는 미용사입니다.');" history.go(-1);</script>";
    exit;
}

$did = mysqli_fetch_row($result_designer)[0];
```

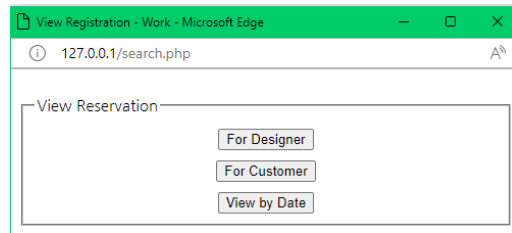
마지막으로 예약 등록을 위해 중복 예약이 존재하는지 확인한 후

```
$sql_datetime = "select * from reservation where did='$did' and datetime='$datetime'";
```

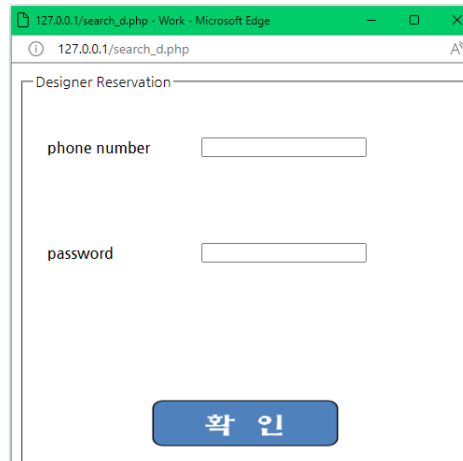
예약을 진행한다.

```
$sql2 = "insert into reservation(cid, did, datetime, sid) values('$cid', '$did', '$datetime', '$sid')";
```

2.4. View Reservation – Designer (Search.php, Search_d.php, Search_d2.php)



포괄적인 예약 조회 메뉴를 위해 search.php 에서 각 버튼 클릭을 통해 해당하는 페이지로 이동한다.



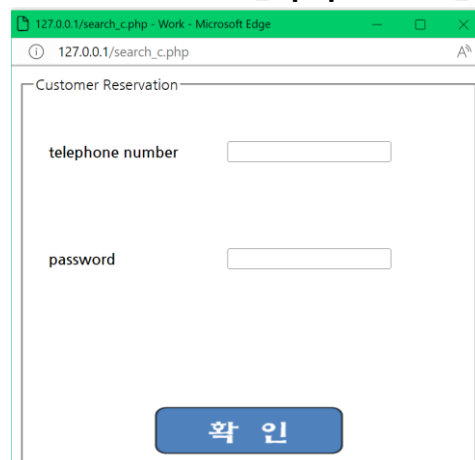
Search_d.php 에서 디자이너 authentication 을 위해 전화번호와 비밀번호를 입력 받아 해당 form 을 search_d2.php 로 제출한다.

Search_d2.php 에서는 이전과 동일한 방식으로 입력되지 않은 변수에 대해 촉구하고 회원 정보를 인증한다. 이후 query 를 통해 디자이너 별 예약 정보를 불러온다.

```
$sql2 = "select distinct datetime, C.fname, C.lname, S.style_name, C.tel_num from reservation, designer D, style S, customer C where D.tel_num='$tel_num'and cid=C.id and did=D.id and sid=S.id" ;
```

불러온 정보를 표 형식으로 출력한다.

2.5. View Reservation – Customer (Search_c.php, Search_c2.php)



고객 별 예약 정보 조회도 2.4와 동일한 작동 방식을 가진다.

Search_c2.php 에서 고객 별 예약 정보를 불러오는 query 문의 차이가 있다.

```
$sql2 = "select distinct datetime, D.fname, S.style_name, D.tel_num from reservation, designer D, style S, customer C where C.tel_num='$tel_num' and did=D.id and cid=C.id and sid=S.id" ;
```

2.6. View Reservation – by Date (Search_date.php, Search_date2.php)

날짜 별 예약 정보(시간, 회원명, 디자이너명, 스타일명)을 조회할 수 있도록 한다.

Search_data.php 는 date 정보를 입력 받아 form 으로 제출한다.

Search_data2.php 는 입력 받은 date 에 대한 query 를 수행하여 필요한 예약 정보를 불러온다.

```
$sql1 = "select distinct TIME(datetime), C.fname, C.lname, D.fname, S.style_name from reservation, designer D, style S, customer C where DATE(datetime)='$date' and cid=C.id and did=D.id and sid=S.id";
```

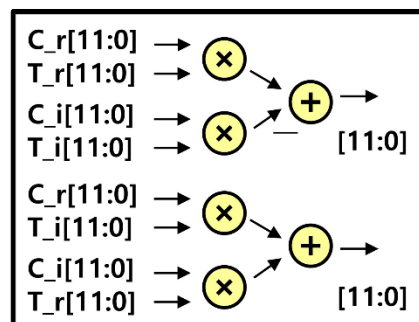
결과는 테이블 형태로 출력한다.

2.7. Customer Cancel Resevation (Cancel.php, Cancel2.php, Cancel3.php)

3. Complex Multiplier (Twiddle Factor)

3.1. Verilog Hardware

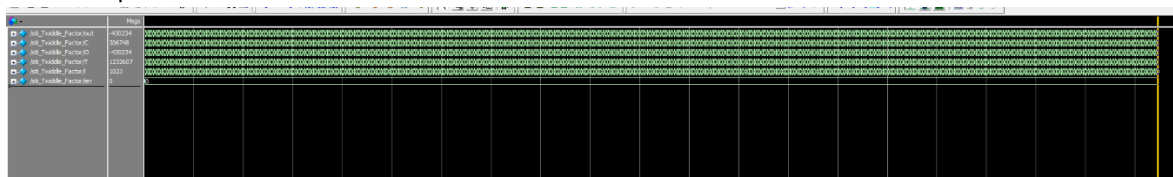
16-bit FFT 설계를 위한 모듈 중, complex multiplier (twiddle factor multiplier)를 설계한다.



Multiplier 의 설계는 위 도면을 따른다. Complex number 에 대한 multiplier 이기에 실수, 허수부를 분리하여 계산한다. 회로 설계는 combinational logic 으로 진행한다.

3.2. Functionality Verification with Testbench Simulation Waveform (Time Diagram)

주어진 input C, Twiddle factor 에 대하여 계산 결과는 주어진 정답 값과 일치함을 확인하였다.



Iteration 의 끝, i=1023 까지 error 가 0 인 것을 확인하였다.

3.3. Synthesis Result

1.3 과 마찬가지로 combinational circuit 의 synthesis 이기에 flipflop 을 추가하지 않았다.

Timing report 는 다음과 같다.

Point	Incr	Path
input external delay	0.00	0.00 r
T[0] (in)	0.00	0.00 r
mult_158/b[0] (Twiddle_Factor_DW_mult_tc_2)	0.00	0.00 r
mult_158/U328/Q (nbis2)	0.36	0.36 r
mult_158/OR_NOT1/Q (and2s1)	0.26	0.62 r
mult_158/U2/Q (hiis1)	0.22	0.84 f
mult_158/U251/Q (mx12is2)	0.34	1.19 r
mult_158/U38/OUTC (haddis2)	0.37	1.55 r
mult_158/U38/OUTS (haddis2)	0.55	2.10 f
mult_158/product[1] (Twiddle_Factor_DW_mult_tc_2)	0.00	2.10 f
add_158/A[1] (Twiddle_Factor_DW01_add_0)	0.00	2.10 f
add_158/U32/Q (oai21s1)	0.23	2.33 r
add_158/U31/Q (oai21s1)	0.30	2.63 f
add_158/U28/Q (oai21s1)	0.23	2.86 r
add_158/U27/Q (oai21s1)	0.30	3.16 f
add_158/U24/Q (oai21s1)	0.23	3.39 r
add_158/U23/Q (oai21s1)	0.30	3.69 f
add_158/U20/Q (oai21s1)	0.23	3.92 r
add_158/U19/Q (oai21s1)	0.30	4.22 f
add_158/U16/Q (oai21s1)	0.23	4.45 r
add_158/U15/Q (oai21s1)	0.32	4.77 f
add_158/U14/Q (oai21s1)	0.26	5.03 r
add_158/U13/Q (hiis1)	0.25	5.29 f
add_158/U12/Q (aoi21s1)	0.16	5.44 r
add_158/U10/Q (hiis1)	0.22	5.66 f
add_158/U9/Q (oai21s1)	0.21	5.87 r
add_158/U8/Q (oai21s1)	0.32	6.19 f
add_158/U7/Q (oai21s1)	0.26	6.45 r
add_158/U6/Q (hiis1)	0.25	6.70 f
add_158/U5/Q (aoi21s1)	0.16	6.86 r
add_158/U3/Q (hiis1)	0.22	7.08 f
add_158/U2/Q (oai21s1)	0.21	7.28 r
add_158/U1/Q (oai21s1)	0.35	7.63 f
add_158/U1_10/OUTC (faddis2)	0.55	8.19 f
add_158/U1_11/OUTC (faddis2)	0.47	8.66 f
add_158/U1_12/OUTC (faddis2)	0.47	9.13 f
add_158/U1_13/OUTC (faddis2)	0.47	9.61 f
add_158/U1_14/OUTC (faddis2)	0.47	10.08 f
add_158/U1_15/OUTC (faddis2)	0.47	10.56 f
add_158/U1_16/OUTC (faddis2)	0.47	11.03 f
add_158/U1_17/OUTC (faddis2)	0.47	11.51 f
add_158/U1_18/OUTC (faddis2)	0.47	11.98 f
add_158/U1_19/OUTC (faddis2)	0.47	12.45 f
add_158/U1_20/OUTC (faddis2)	0.47	12.93 f
add_158/U1_21/OUTS (faddis2)	0.76	13.69 r
add_158/SUM[21] (Twiddle_Factor_DW01_add_0)	0.00	13.69 r
out[11] (out)	0.00	13.69 r
data arrival time		13.69
(Path is unconstrained)		

1.3 과 동일한 이유로 invalid timing report 임이 확인된다.

Area report 는 다음과 같다.

```
Library(s) Used:
    lec25dsc25_SS (File: /home/admin/lib/lec25/lec25dsc25_SS.db)

Number of ports:      72
Number of nets:      162
Number of cells:      6
Number of references: 6

Combinational area:    108648.346741
Noncombinational area: 0.000000
Net Interconnect area: undefined (No wire load specified)

Total cell area:      108648.343750
Total area:           undefined
```

Total area 는 108648.343750nm^2 이다.

Power report 는 다음과 같다.

```

Global Operating Voltage = 2.25
Power-specific unit information :
Voltage Units = 1V
Capacitance Units = 1.000000pf
Time Units = 1ns
Dynamic Power Units = 1mW      (derived from V,C,T units)
Leakage Power Units = 1pW

Cell Internal Power = 162.9696 mW   (64%)
Net Switching Power = 90.3992 mW   (36%)
-----
Total Dynamic Power  = 253.3689 mW (100%)
Cell Leakage Power   = 1.2053 mW

```

소모 전력은 253.3689 mW 이다.

Compile 된 schematic 과 critical path 는 다음과 같다.

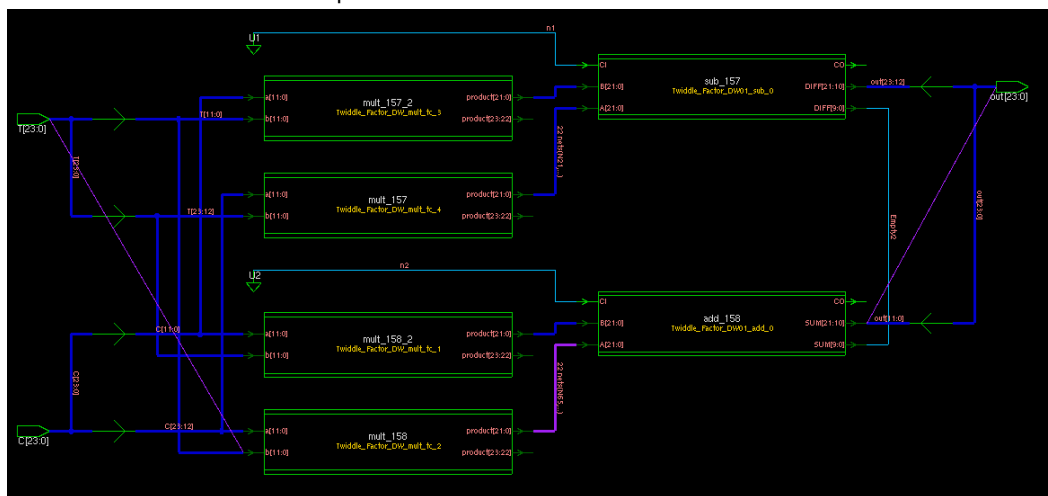


Figure 1 전체 schematic

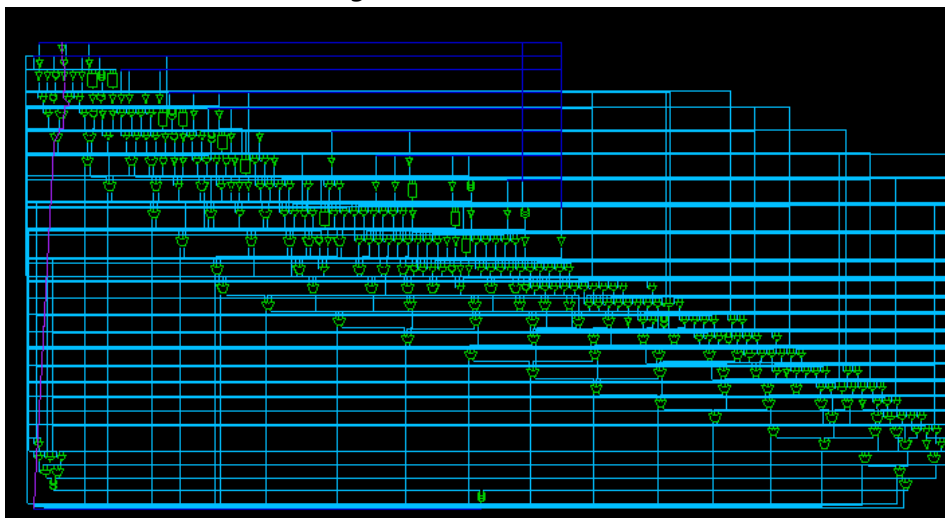


Figure 2 multiply module 내부

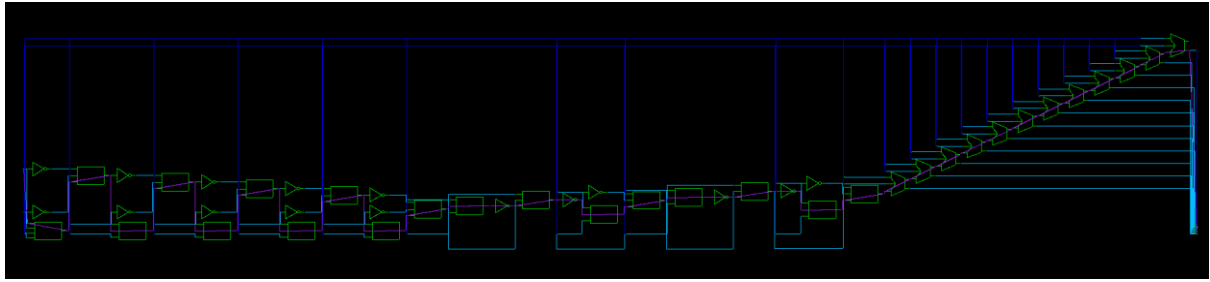


Figure 3 add module 내부

Critical path 는 2.1. 도면의 한 path, multiply 이후 add 를 거치는 것을 확인하였다.

Reference

[1] Verilog HDL: A Guide to Digital Design and Synthesis