

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. Some nodes are highlighted with blue circles, and others with blue dots. The diagram is rendered in a light gray color.

作業五 : Average

2022/05/16



設計規格描述

• 請用設計出一個取平均值之電路-Average

▸ 輸入/輸出腳位

訊號名稱	I/O	Bit數	描述
clk	I	1	本電路同步於時脈正緣之同步設計
reset	I	1	高準位非同步(active high asynchronous)之電路重置訊號
data	I	8	八位元無號數輸入
valid	O	1	有效輸出資料之通知訊號 當為 High 時，表示目前輸出out為有效輸出
out	O	8	八位元無符號平均值輸出

▸ 提供檔案

檔案名稱	描述
Average.v	Verilog檔案(自行設計)，用於RTL模擬。注意：I/O 腳位已定義，請勿更改！
testfixture.v	Testbench(測試檔)，用於RTL模擬、Gate-Level模擬
pattern.dat	作為電路模擬時，輸入訊號測試資料 一同放入ModelSim專案內
golden.dat	作為電路模擬時，比對輸出結果的資料 一同放入ModelSim專案內

功能描述

- 資料輸入:當reset訊號送出後，data線會在每個clk開始給值，並在128個clk後結束輸入。組成16x8陣列
- 資料輸出:運算出平均值(四捨五入)，後將結果在每個clk輸出至out線，並且把valid拉成High

	0	1	2	3	4	5	6	7
0	255	49	126	146	146	198	101	217
1	255	242	138	211	28	84	116	60
2	115	151	242	175	29	49	115	222
3	137	198	208	126	95	154	8	182
4	62	249	40	254	88	20	218	31
5	130	94	113	168	246	233	228	158
6	100	222	156	179	123	122	129	188
7	143	189	41	197	139	45	149	124
8	215	216	101	56	75	0	224	119
9	241	138	203	2	166	185	101	148
10	76	43	125	65	184	185	254	98
11	254	47	228	96	74	107	0	13
12	224	236	106	229	127	86	19	251
13	191	247	90	214	150	206	19	7
14	178	37	218	197	203	66	40	82
15	145	48	11	116	101	52	235	196

pattern.dat



	0	1	2	3	4	5	6	7
0	255	146	132	179	87	141	109	139
1	185	197	190	193	29	67	116	141
2	126	175	225	151	62	102	62	202
3	100	224	124	190	92	87	113	107
4	96	172	77	211	167	127	223	95
5	115	158	135	174	185	178	179	173
6	122	206	99	188	131	84	139	156
7	179	203	71	127	107	23	187	122
8	228	177	152	29	121	93	163	134
9	159	91	164	34	175	185	178	123
10	165	45	177	81	129	146	127	56
11	239	142	167	163	101	97	10	132
12	208	242	98	222	139	146	19	129
13	185	142	154	206	177	136	30	45
14	162	43	115	157	152	59	138	139

golden.dat

功能描述

	0	1	2	3	4	5	6	7
0	255	49	126	146	146	198	101	217
1	255	242	138	211	28	84	116	60
2	115	151	242	175	29	49	115	222
3	137	198	208	126	95	154	8	182
4	62	249	40	254	88	20	218	31
5	130	94	113	168	246	233	228	158
6	100	222	156	179	123	122	129	188
7	143	189	41	197	139	45	149	124
8	215	216	101	56	75	0	224	119
9	241	138	203	2	166	185	101	148
10	76	43	125	65	184	185	254	98
11	254	47	228	96	74	107	0	13
12	224	236	106	229	127	86	19	251
13	191	247	90	214	150	206	19	7
14	178	37	218	197	203	66	40	82
15	145	48	11	116	101	52	235	196

pattern.dat



	0	1	2	3	4	5	6	7
0	255	146	132	179	87	141	109	139
1	185	197	190	193	29	67	116	141
2	126	175	225	151	62	102	62	202
3	100	224	124	190	92	87	113	107
4	96	172	77	211	167	127	223	95
5	115	158	135	174	185	178	179	173
6	122	206	99	188	131	84	139	156
7	179	203	71	127	107	23	187	122
8	228	177	152	29	121	93	163	134
9	159	91	164	34	175	185	178	123
10	165	45	177	81	129	146	127	56
11	239	142	167	163	101	97	10	132
12	208	242	98	222	139	146	19	129
13	185	142	154	206	177	136	30	45
14	162	43	115	157	152	59	138	139

golden.dat

注意事項

- bit數擴展: $(49 + 242) = 291$ 。但由於只有8bit，當相加結果大於255時會產生溢位，所以只會得到35

bit	8	7	6	5	4	3	2	1	0
10進制	291								
2進制	1	0	0	1	0	0	0	1	1



bit	7	6	5	4	3	2	1	0
10進制	35							
2進制	0	0	1	0	0	0	1	1

✗ 錯誤

範例寫法

`reg [7:0] a, b;`

`wire [8:0] sum;`

`assign sum = a + b;`

- 四捨五入: $291 / 2 = 145.5$

bit	7	6	5	4	3	2	1	0
10進制	145							
2進制	1	0	0	1	0	0	0	1

1

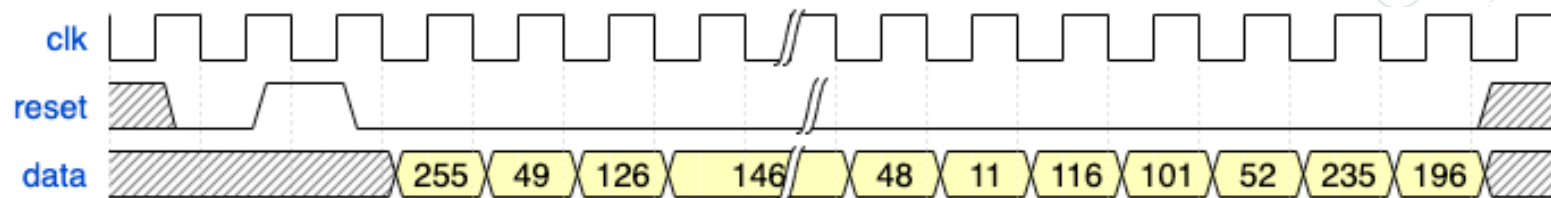
範例寫法

`out = (sum >> 1) + sum[0];`

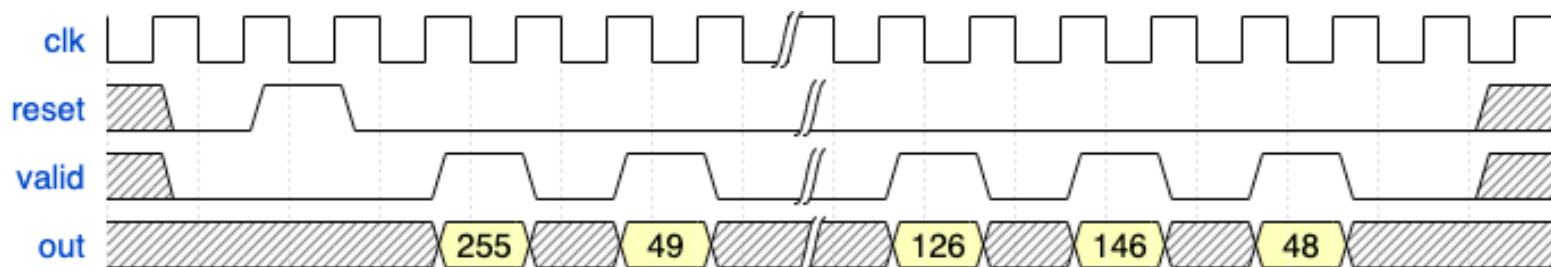
`//or`

`out = sum[8:1] + sum[0];`

電路波形範例



▲資料輸入波形參考



▲結果輸出波形參考

解題方法一

- 步驟一: 先花128個clk將整個陣列存起來

☐ reg宣告: (1) 8位元, 128大小的一維陣列、(2) 8位元, 16 x 8大小的二維陣列

- 步驟二: 運算

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71
72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87
88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103
104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127

[7:0] a [127:0]

	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	0	1	2	3	4	5	6	7
2	0	1	2	3	4	5	6	7
3	0	1	2	3	4	5	6	7
4	0	1	2	3	4	5	6	7
5	0	1	2	3	4	5	6	7
6	0	1	2	3	4	5	6	7
7	0	1	2	3	4	5	6	7
8	0	1	2	3	4	5	6	7
9	0	1	2	3	4	5	6	7
10	0	1	2	3	4	5	6	7
11	0	1	2	3	4	5	6	7
12	0	1	2	3	4	5	6	7
13	0	1	2	3	4	5	6	7
14	0	1	2	3	4	5	6	7
15	0	1	2	3	4	5	6	7

[7:0] b [15:0][7:0]

解題方法二

• Line Buffer

[?] reg宣告: 8位元, 8大小的一維陣列(例: reg [7:0] buffer [7:0])

運作流程

clk	buffer								data
reset 後	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	
0	0	0	0	0	0	0	0	0	255
1	0	0	0	0	0	0	0	255	49
2	0	0	0	0	0	0	255	49	126
3	0	0	0	0	0	255	49	126	146
4	0	0	0	0	255	49	126	146	146
5	0	0	0	255	49	126	146	146	198
6	0	0	255	49	126	146	146	198	101
7	0	255	49	126	146	146	198	101	217
8	255	49	126	146	146	198	101	217	255 valid拉high
9	49	126	146	146	198	101	217	242	242

pattern資料輸入

255	49	126	146	146	198	101	217
255	242					

範例寫法

```
integer i;
```

```
buffer [7] <= data;
```

```
for(i=0;i<7;i=i+1)begin
```


```
    data[i] <= data[i+1];
```

```
end
```


電路模擬範例

- RTL、Gate-Level模擬 -> PASS

► **Modelsim模擬步驟、Quartus合成步驟請參考tools教學**

```
#####  
**                                     **  
**   Congratulations !!             **  
**                                     **  
**   Simulation PASS!!              **  
**                                     **  
#####  
  
#####  
# ** Note: $finish :  
# Time: 3275 ns Iteration: 2 Instance: /testfixture  
# 1
```

助教只檢查Gate-Level模擬 是否通過

上傳檔案

- **RTL File**

1. Average.v (你所有的Verilog RTL Code)

- **Gate-Level File**

1. Average.vo (Quartus 生成的Gate-Level Netlist)
2. Average.sdo (Quartus 生成的 SDF 時序資訊)

- **Document**

1. report.pdf (你的設計報告文件 **請轉成PDF格式**)

▲請包成zip格式，並命名為 學號__HW5.zip

上傳檔案注意事項

- 1.Quartus 請一律使用 **Cyclonell** 晶片
- 2.**請勿**將整個專案資料夾包進zip檔上傳，如發現上傳整個專案者一律**視同未繳交**
- 3.設計**未完成的**檔案請勿上傳
- 4.上傳後請再次檢查上傳是否有誤，如上傳錯誤一律**視同未繳交**
- 5.補交作業一律在**5/30**當天補交