[2010년 디지털 시스템 설계 및 실험 기말고사 2 답안지]

출제: 채수익

1.

(a) (5 pts)

$x_i y_i$	00	01	01	$0\bar{1}$	$0\bar{1}$	11	$1\bar{1}$	$\bar{1}\bar{1}$
$x_{i-1} y_{i-1}$	-	neither is $\bar{1}$	at least one is $\bar{1}$	neither is $\bar{1}$	at least one is $\bar{1}$	-	-	-
c_i	0	1	0	0	$\bar{1}$	1	0	$\bar{1}$
u_i	0	$\bar{1}$	1	$\bar{1}$	1	0	0	0

(단, 10인 case나 $\bar{1}0$인 case는 위의 경우에서 둘을 바꾼 경우와 동일하므로 생략한다)

위와 같이 각 bit position에 대해 c_i, u_i를 구한 다음, 아래과 같이 sum을 구한다.

$$s_i = u_i + c_{i-1}$$

위의 table을 맞게 construct한 경우 (4 pts)

- 단순히 binary SD number system을 설명한 경우 (1 pts)
- 위의 table 대신 아래의 식을 쓴 경우 (4 pts)

$$c_i = \begin{cases} 1, & \text{if } (x_i + y_i) \geq 1 \\ \bar{1}, & \text{if } (x_i + y_i) \leq \bar{1} \\ 0, & \text{if } (x_i + y_i) = 0 \end{cases}$$
$$u_i = x_i + y_i - 2c_i$$

sum을 구하는 식을 맞게 썼을 경우 (1 pts)

- 첨자를 표시하지 않은 경우 (0.5 pts)

(b) (10 pts)

만일 $\bar{1} \leq s_i \leq 1$이 만족된다면 carry propagation이 없다.

00, 11, $1\bar{1}, \bar{1}\bar{1}$인 경우에는 u_i가 0이므로 항상 조건을 만족한다.

01인 경우에는, 두 경우로 나누어 생각한다.

(1) neither of $x_{i-1}y_{i-1}$ is $\bar{1}$

이 경우에 u_i는 $\bar{1}$이므로 c_{i-1}이 $\bar{1}$가 아니면 조건을 만족한다. 그런데 $x_{i-1}y_{i-1}$가 둘 다 $\bar{1}$가 아니므로 c_{i-1}이 $\bar{1}$가 될 수 없다.

(2) at least one of $x_{i-1}y_{i-1}$ is $\bar{1}$

이 경우에 u_i는 1이므로 c_{i-1}이 1이 아니면 조건을 만족한다. 그런데 $x_{i-1}y_{i-1}$ 둘 중 하나가 $\bar{1}$가 이므로 c_{i-1}이 1가 될 수 없다.

$0\bar{1}$인 경우에도, 두 경우로 나누어 생각한다.

(1) neither is $\bar{1}$

이 경우에 u_i는 $\bar{1}$이므로 c_{i-1}이 $\bar{1}$가 아니면 조건을 만족한다. 그런데 $x_{i-1}y_{i-1}$가 둘 다 $\bar{1}$가 아니므로 c_{i-1}이 $\bar{1}$가 될 수 없다.

(2) at least one is $\bar{1}$

이 경우에 u_i는 1이므로 c_{i-1}이 1이 아니면 조건을 만족한다. 그런데 $x_{i-1}y_{i-1}$ 둘 중 하나가 $\bar{1}$가

이므로 c_{i-1}이 1가 될 수 없다.

Carry propagation이 없는 조건을 쓴 경우 (3 pts)

각 case에 대해 해당 조건이 만족됨을 보인 경우 (각각 1 pts씩)

- 만일 (a)에서 table 대신 수식을 쓴 경우, carry propagation이 발생하므로 carry propagation이 없는 경우에 대해서만 설명한 경우 (2 pts)
- (a)에서 수식을 썼는데, carry propagation이 없다고 한 경우 (0 pts)

앞의 (a)를 정확히 썼는데, 단순히 말로 carry propagation이 없다고 풀어쓴 경우 (5 pts)

2.

(a) (5 + 5 pts)

							$-a_7$	a_6	a_5	a_4	a_3	a_2	a_1	a_0
						$\times$	$-x_7$	x_6	x_5	x_4	x_3	x_2	x_1	x_0
						$+1$	$\overline{a_7x_0}$	a_6x_0	a_5x_0	a_4x_0	a_3x_0	a_2x_0	a_1x_0	a_0x_0
						$\overline{a_7x_1}$	a_6x_1	a_5x_1	a_4x_1	a_3x_1	a_2x_1	a_1x_1	a_0x_1	
					$\overline{a_7x_2}$	a_6x_2	a_5x_2	a_4x_2	a_3x_2	a_2x_2	a_1x_2	a_0x_2		
$+$				$\overline{a_7x_3}$	a_6x_3	a_5x_3	a_4x_3	a_3x_3	a_2x_3	a_1x_3	a_0x_3			
			$\overline{a_7x_4}$	a_6x_4	a_5x_4	a_4x_4	a_3x_4	a_2x_4	a_1x_4	a_0x_4				
		$\overline{a_7x_5}$	a_6x_5	a_5x_5	a_4x_5	a_3x_5	a_2x_5	a_1x_5	a_0x_5					
	$\overline{a_7x_6}$	a_6x_6	a_5x_6	a_4x_6	a_3x_6	a_2x_6	a_1x_6	a_0x_6						
-1	a_7x_7	$\overline{a_6x_7}$	$\overline{a_5x_7}$	$\overline{a_4x_7}$	$\overline{a_3x_7}$	$\overline{a_2x_7}$	$\overline{a_1x_7}$	$\overline{a_0x_7}$						
$-p_{15}$	p_{14}	p_{13}	p_{12}	p_{11}	p_{10}	p_9	p_8	p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0

위와 같은 그림을 그리거나 혹은 말로 풀어쓰고, 이를 설명한다.

(1) 과정

왜 +1, -1, inversion 등을 사용하는지에 대한 설명 (5 pts)

- 단순히 multiplication에 대한 설명만 있는 경우 (2 pts)

(2) 결과

위의 그림을 정확히 그린 경우 (MSB의 -1을 1로 쓰고 2's complement라는 말을 한 경우 포함) (5 pts)

- 일부 잘못된 경우 (2 pts)

(b) (5 pts)

최대 8개의 partial product가 존재한다. 한 단계를 거치면 4개로 줄일 수 있고, 두 단계를 거치면 2개로 만들 수 있다. 즉, 두 단계가 필요하다.

두 단계를 맞춘 경우 (5 pts)

틀린 경우 (0 pts)

(c) (5 pts)

(a)의 그림을 확인하면 output이 16-bit이다. 일반적으로 8-bit 2개를 곱하면 16-bit 결과가 나오므로 16-bit adder가 필요하다. 단 LSB의 경우, partial product가 하나이므로 굳이 더할 필요가 없으므로 15-bit adder를 써도 된다.

16-bit, 혹은 15-bit이라고 쓴 경우 (5 pts)

- 그 외 15-bit보다 작은 bit을 쓴 경우 (2 pts)

(d) (5 + 5 pts)

Carry-lookahead adder, Carry-skip adder, Conditional-sum adder 등 log n의 속도를 낼 수 있는 adder 중 하나를 block diagram으로 그리고, 해당 adder의 속도를 제시한다.

Block diagram을 정확히 그린 경우 (5 pts)

- 만일 block diagram 대신 verilog code를 작성한 경우 (5 pts)

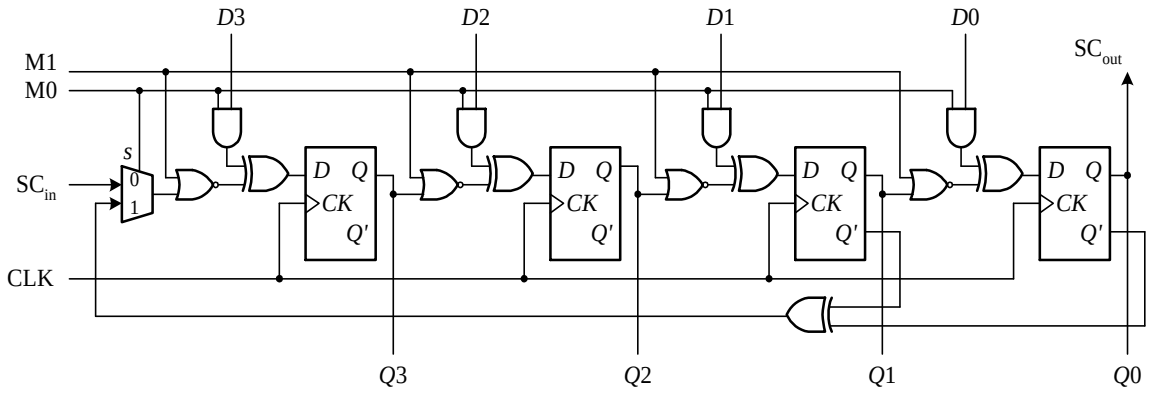
해당 adder의 속도를 제시한 경우 (Gate delay를 가정하고 속도를 이에 따라 나타내면 된다) (5 pts)

- 단순히 log N adder를 쓴다고 한 경우 (3 pts)

RCA를 쓴 경우 (총 2 pts)

3.

(a) (5 pts)



(a) Logic diagram

M1	M0	Function
0	0	Scan mode
0	1	MISR
1	0	Clear register
1	1	Parallel load

(b) Function selection

Scan mode : scan-path method를 지원한다.

MISR (Multiple-input signature register) : signature analysis를 지원한다.

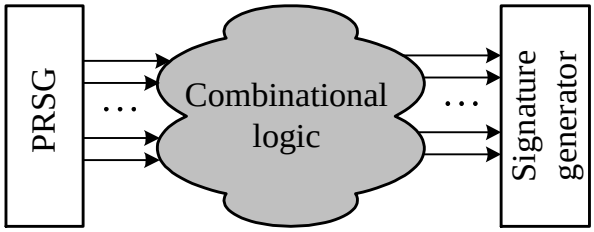
Clear mode : register의 내용을 clear한다.

Parallel-load mode : 각 register에 parallel load를 수행한다.

Logic diagram을 정확히 그린 경우 (M0, M1의 경우 명확히 표현해야 한다) (2 pts)

Function selection에 대해 정확히 나타낸 경우 (1 pts)
각 mode에 대한 설명 (각 0.5 pts씩)

(b) (5 pts)



(c) Application example

In the normal operation mode, the PRSG (PR-sequence generator) at the input of the combinational logic circuit is **isolated and effectively removed by a mux**. The registers of the signature generator are used as the state registers for the combinational logic circuit. **In the test mode**, PRSG generates **random test vectors** for combinational logic and signature generator performs the **signature analysis**.

PRSG, combinational logic, signature generator로 구성된 그림 (2 pts)
Normal mode와 test mode를 구분하여 설명 (1 pts)
각각의 mode에서 PRSG, signature generator가 어떻게 사용되는지 설명 (각 1 pts)

4. (2 + 2 + 2 + 2 + 2 pts)

(a) fault model
Fault model is an **engineering model** of the physical **defect in a circuit**.
(b) fault coverage
Fault coverage is defined as **the number of detected faults divided by total number of faults**.

$$\text{fault coverage} = \frac{\text{number of detected faults}}{\text{total number of faults}}$$

(c) slack time
Slack time is defined as the **subtraction** of **the arrival time from the required time**.

$$\text{slack time} = \text{required time} - \text{arrival time}$$

(d) structural coverage
Structural coverage indicates which **key parts of the HDL code structure** have been **covered**.
(e) functional coverage
Functional coverage makes sure that **all possible legal values of input stimuli** are exercised in **all possible combinations at all possible times**.

밑줄 친 부분의 내용이 들어가야 함 (각 1 pts씩)
- (e)의 경우, 해당 회로가 기능을 제대로 수행하는지를 확인한다고 써도 (2 pts)
- 그 외 다른 방식으로 설명한 경우 맞으면 (2 pts) 일부 내용이 틀리면 (1 pts) 의미가 불분명한 경우 (0 pts)

5.

(a) (5 pts)

False paths, multi-cycle paths

둘 다 맞았을 경우 (5 pts)

- 오기라고 생각되는 경우에도 (5 pts) (ex: false path -> fault path)

하나만 맞았을 경우 (2 pts)

(b) (10 pts)

False path : a timing path that **does not propagate a signal** Multi-cycle path : a timing path where **data takes more than one cycle** to reach its destination

밑줄 친 부분의 내용이 들어가야 함 (각 5 pts)

틀릴 경우 (0 pts)

6.

(a) (2 pts)

$x_i \frac{df(X)}{dx_i} = 1$ 이 되는 input test vector를 찾는다.

수식을 정확히 쓴 경우에만 (2 pts)

- 만일 위의 수식을 만족하지 않는 경우를 찾는다고 쓴 경우 (1 pts)

(b) (2 pts)

$x_i' \frac{df(X)}{dx_i} = 1$ 이 되는 input test vector를 찾는다.

수식을 정확히 쓴 경우에만 (2 pts)

- 만일 위의 수식을 만족하지 않는 경우를 찾는다고 쓴 경우 (1 pts)

(c) (6 pts)

f를 node로 사용하므로, F(X)로 회로의 logic function을 써서 뜻을 명확히 한다.

$$f = bc$$

$$F(X, f) = a + f' + d + e$$

$$\frac{dF}{df} = (a + 1 + d + e) \oplus (a + 0 + d + e)$$
$$= (a + d + e)' = a'd'e'$$

따라서,

$$f' \frac{dF}{df} = (b' + c')a'd'e' = a'b'd'e' + a'c'd'e'$$

이로부터 다음과 같은 test vector set을 얻는다.

$$\{(0, 0, \emptyset, 0, 0), (0, \emptyset, 0, 0, 0)\}$$

f를 정확히 f = bc로 표현한 경우 (1 pts)

$\frac{dF}{df}$를 정확히 구한 경우 (2 pts)

test vector set을 정확히 구한 경우 (3 pts)

7.

(a) (2 pts)

x	y	z	Faults covered
0	0	D	z/0
0	1	D	x/1, z/0
1	0	D	y/1, z/0
1	1	D′	x/0, y/0, z/1

정확히 맞을 경우 (faults covered는 관계없음) (2 pts)

하나라도 틀린 경우 (0 pts)

(b) (2 pts)

x	y	z
D	0	D′
0	D	D′
D′	0	D
0	D′	D

정확히 맞을 경우 (2 pts)

하나라도 틀린 경우 (0 pts)

(c) (2 pts)

x	y	z
0	Ø	0
Ø	0	0
1	1	1

정확히 맞을 경우 (2 pts)

- sc가 아닌 AND의 truth table을 단순히 그렸을 경우 (1 pts)

그 외의 경우 (0 pts)

(d) (6 pts)

(1) fault sensitization

select a pdcf for the fault for which a test pattern is to be generated

(2) D-drive

use pdcs to propagate the D signal to at least one primary output of the circuit

(3) Consistency operations

use the sc of each logic module to perform the consistency operation

각 단계에 대한 설명 (pdcf, pdc, sc에 대한 내용이 반드시 있어야 함) (각 2 pts씩)

- pdcf, pdc, sc에 대한 내용은 빠졌으나, 내용은 맞을 경우 (1 pts씩)

(e) (6 pts)

a	b	c	d	e	f	g	h	i
1							0	D
					1	1	0	
	1	1			1			
			0	0		1		

첫째 줄이 fault sensitization에 해당한다.

a와 연결된 OR gate의 output이 primary output에 해당하므로 D-drive는 아무런 변화를 가져오지 않는다.

둘째 줄부터 넷째 줄은 consistency operation에 해당한다.

따라서 test vector는 {(1, 1, 1, 0, 0)}에 해당한다.

---

fault sensitization에 해당하는 부분, 즉 pdcf를 정확히 선택한다 (2 pts)

consistency operation에 해당하는 부분 (2 pts)

- D-drive를 진행한 중간 결과가 있을 경우 (-1 pts)

결과 test vector를 정확히 제시한 경우 (2 pts)

8.

(a) (4 pts)

Fault simulation is to select a primary input combination and determine its fault detection capabilities by simulation. For each combination of inputs, (1)**both fault-free and faulty circuits are simulated in parallel and their outputs are compared**. If two circuits produce **different outputs, the input combination becomes the test vector**. **Simulation is repeated until** all faults are covered, at least an acceptable number of faults are covered, or some predefined stopping point is reached.

(1)은 반드시 있어야 함 (2 pts)

언제 input combination이 test vector가 되는지 써야 함 (1 pts)

repeat until의 조건은 세 가지 중 하나라도 쓰면 (1 pts)

(b) (6 pts)

x	y	z	f	f_{x0}	f_{x1}	f_{y0}	f_{y1}	f_{z0}	f_{z1}	f_{a0}	f_{a1}	f_{f0}	f_{f1}	Faults detected
0	0	0	1	1	1	*	1	*	**0**	1	**0**	**0**	1	z1, a1, f0
0	0	1	0	0	0	*	0	*	*	0	*	*	**1**	f1
0	1	0	1	1	**0**	1	1	*	*	1	*	*	*	x1
0	1	1	0	0	*	0	0	*	*	0	*	*	*	
1	0	0	1	1	*	1	**0**	1	*	1	*	*	*	y1
1	0	1	0	0	*	0	*	**1**	0	0	*	*	*	z0
1	1	0	0	**1**	0	**1**	0	*	0	**1**	0	*	*	x0, y0, a0
1	1	1	0	*	0	*	0	*	0	*	0	*	*	

(*은 해당 input pattern에 대해 simulation이 수행되지 않았음을 의미한다)

(밑줄 친 부분이 column method에 의해 발견된 fault에 해당한다)

test vector는 **{(0,0,0), (0,0,1), (0,1,0), (1,0,0}, (1,0,1), (1,1,0)}** 가 된다.

Table을 위의 형태로 construct한 경우 (1 pts)

column method를 정확하게 진행한 경우 (2 pts)

- reduction을 수행한 경우, reduction에 대해서 column method를 정확하게 진행해야 한다.
- column을 위와 다르게 배열한 경우에도 column method를 정확히 진행해야 한다.

test vector를 정확히 구한 경우 (3 pts)

- 3개 이상의 test vector를 정확히 구한 경우 (2 pts)
- reduction을 수행한 경우, 이에 맞게 test vector를 구하면 된다