디지털논리회로1 4차 과제

40-Bit Ripple Carry Adder

2020202031

김재현

목차

-Problem statement

ripple carry adder는 1bit adder를 여러 개 이어 붙여서 만든 2진수 덧셈 연산기입니다. 1bit adder로 full adder를 사용하여 full adder의 C_{out}이 다음 full adder의 C_{in}으로 입력되도록 이어 붙였습니다.

-Descriptions for inputs and outputs

rca_40b 모듈의 입력 값으로는 A, B, Cin이 있습니다. 출력 값으로는 S, Cout이 있습니다. 40비트 2진수 덧셈을 구현해야 하므로 입력 값 A, B와 출력 값 S는 40비트를 저장할 수 있는 bus로 선언했습니다. 그리고 초기에 RCA로 입력되는 Cin과 최종적으로 출력되는 Cout은 1비트로 선언했습니다.

-Gate-level design and schematic of a full adder

full adder truth table

C_{in}	A	B	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

S에 대한 K-map 작성

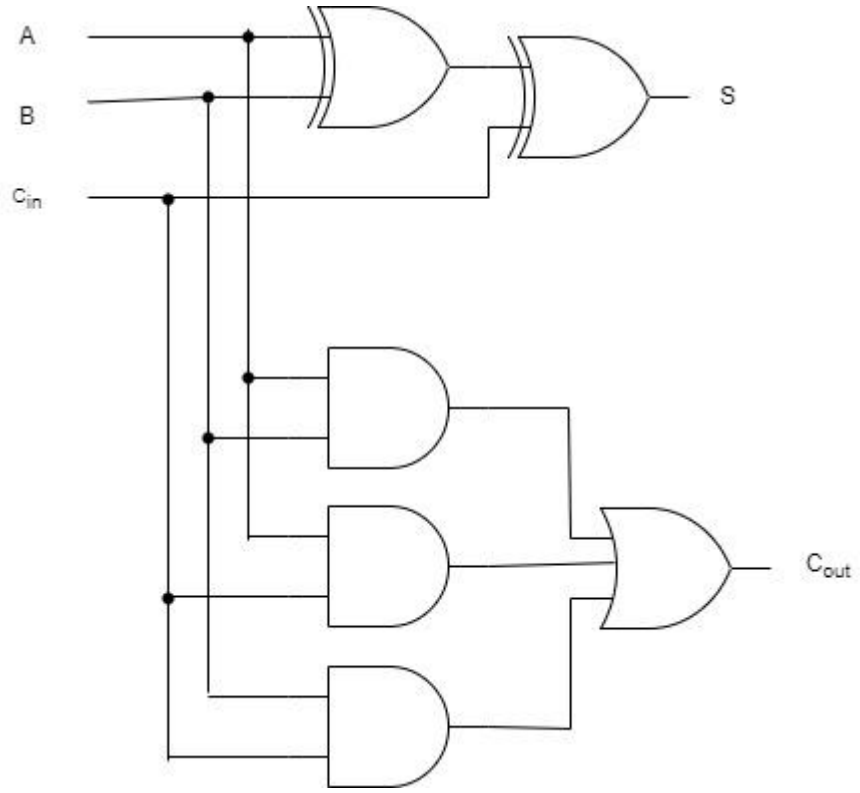
AB \ C_{in}	0	1
00	0	(1)
01	(1)	0
11	0	(1)
10	(1)	0

$S = A'B'C_{in} + A'BC_{in}' + ABC_{in} + AB'C_{in}'$

C_{out}에 대한 K-map 작성

AB \ C_{in}	0	1
00	0	0
01	0	1
11	1	1
10	0	1

$C_{out} = AB + BC_{in} + AC_{in}$


A
B
C_{in}
S
C_{out}

-Block diagram of RCA code

-Descriptions of the operation of RCA code

1) A=40'b0000000000000000000000000000000000000000
B=40'b0000000000000000000000000000000000000000

A와 B가 0이기 때문에 덧셈 결과 역시 0이다. 이 때 Cin의 값에 상관 없이 오버플로우는 일어나지 않는다. 덧셈 결과가 1이 차이 날 뿐이다.

2) A=40'b1111100000111110000011111000001111100000
B=40'b0000011111000001111100000111110000011111

A와 B의 모든 비트가 다 다르기 때문에 Cin이 0일 때는 덧셈의 결과가 40비트로 표현 가능한 최대의 값이 된다. 하지만 Cin이 1이라면 오버플로우가 발생해 덧셈의 결과가 0이 된다.

3) A=40'b1111111111111111111111111111111111111111
B=40'b1111111111111111111111111111111111111111

A와 B가 40비트로 표현 가능한 최댓값이다. Cin의 값에 상관 없이 항상 오버플로우가 일어나는 경우다. 이 역시도 Cin으로 인해 달라지는 점은 덧셈 결과가 1이 차이 난다는 것이다.

이렇듯 가장 작은 자리의 덧셈연산에 포함되어 더해지는 Cin은 단순히 덧셈 결과에 차이를 만들어낼 수 있다는 것 외에도 오버플로우 발생 여부에도 영향을 미칠 수 있습니다. 따라서 Cin은 RCA에서 반드시 고려해야 할 필수적인 요소입니다.

-Code analysis

module rca_40b(S, A, B, Cout, Cin);

rca_40b라는 이름의 40-bit RCA를 선언. 매개변수로는 S, A, B, Cout, Cin이 존재함

input [39:0] A, B;

A, B를 40비트 크기의 버스 입력으로 선언

input Cin;

Cin을 1비트 크기의 입력으로 선언

output [39:0] S;

S를 40비트 크기의 버스 출력으로 선언

outputCout;

Cout을 1비트 크기의 출력으로 선언

wire [39:0] Carry; // Cout save

Cout을 저장할 40비트 버스 Carry 선언

full_adder fa1(A[0], B[0], Cin, Carry[0], S[0]);

full adder 입력으로 A, B의 마지막 비트, Cin을 입력했고, 출력으로 Carry와 S의 마지막 비트를 입력.

genvar i;

loop index 선언

generate

loop간 인스턴스들 간의 충돌을 방지

for(i=1;i<40;i=i+1) begin: rca_loop

full_adder fa2(A[i], B[i], Carry[i-1], Carry[i], S[i]);

end

A, B의 모든 비트와 이전 루프에서 출력된 Carry(=Cout)을 full adder에 입력하고 그 결과값을 Carry, S에 저장함.

assign Cout=Carry[39];

rca_40b의 출력값 Cout에 Carry의 MSB를 assign함.

endgenerate

generate의 끝

endmodule

모듈의 끝

module full_adder(input A, B, Cin, output Cout, Sum);

full_adder라는 이름의 full adder를 선언. 매개변수로는 입력 값 A, B, Cin 출력 값 Cout, Sum이 존재함.

assign w1 = A ^ B;

A, B 의 XOR 연산 결과를 w1 에 assign

assign w2 = A & B;

A, B 의 AND 연산 결과를 w2 에 assign

assign w3 = A & Cin;

A, Cin 의 AND 연산 결과를 w3 에 assign

assign w4 = B & Cin;

B, Cin 의 AND 연산 결과를 w4 에 assign

assign Cout = w2 | w3 | w4;

w2, w3, w4 의 OR 연산 결과를 Cout 에 assign

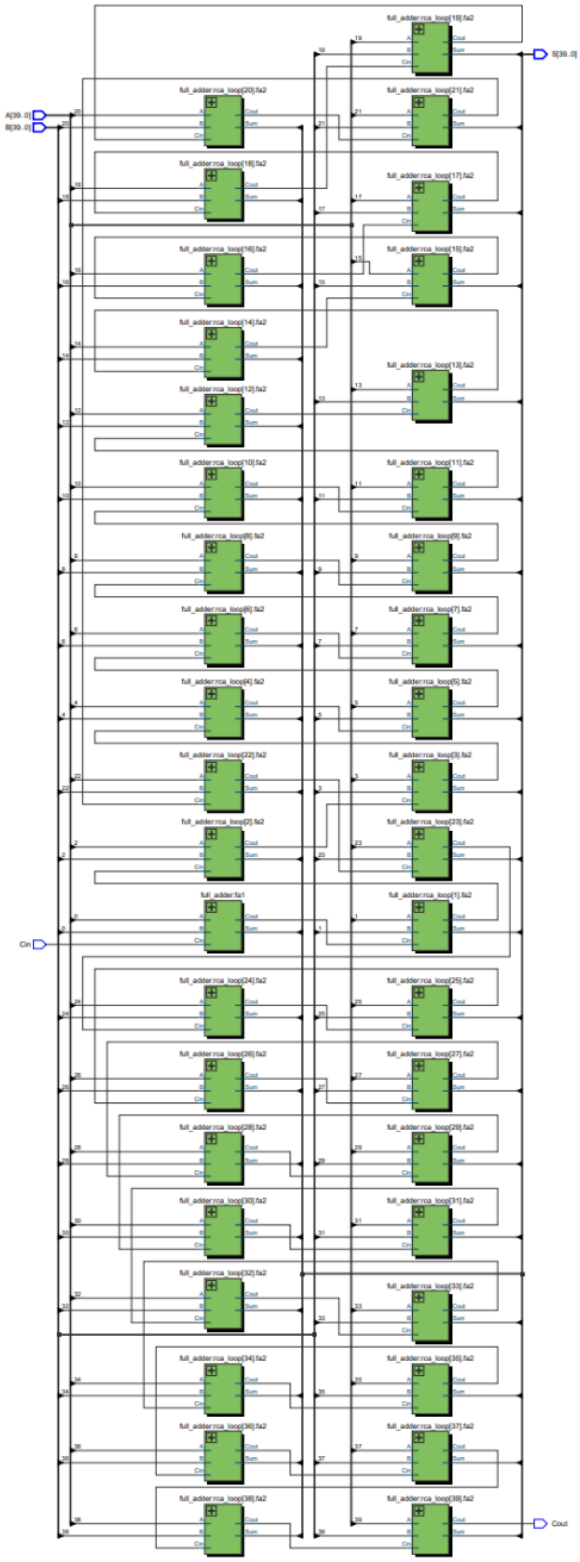
assign Sum = w1 ^ Cin;

w1, Cin 의 OR 연산 결과를 Sum 에 assign

endmodule

모듈의 끝

RTL view and flow summary

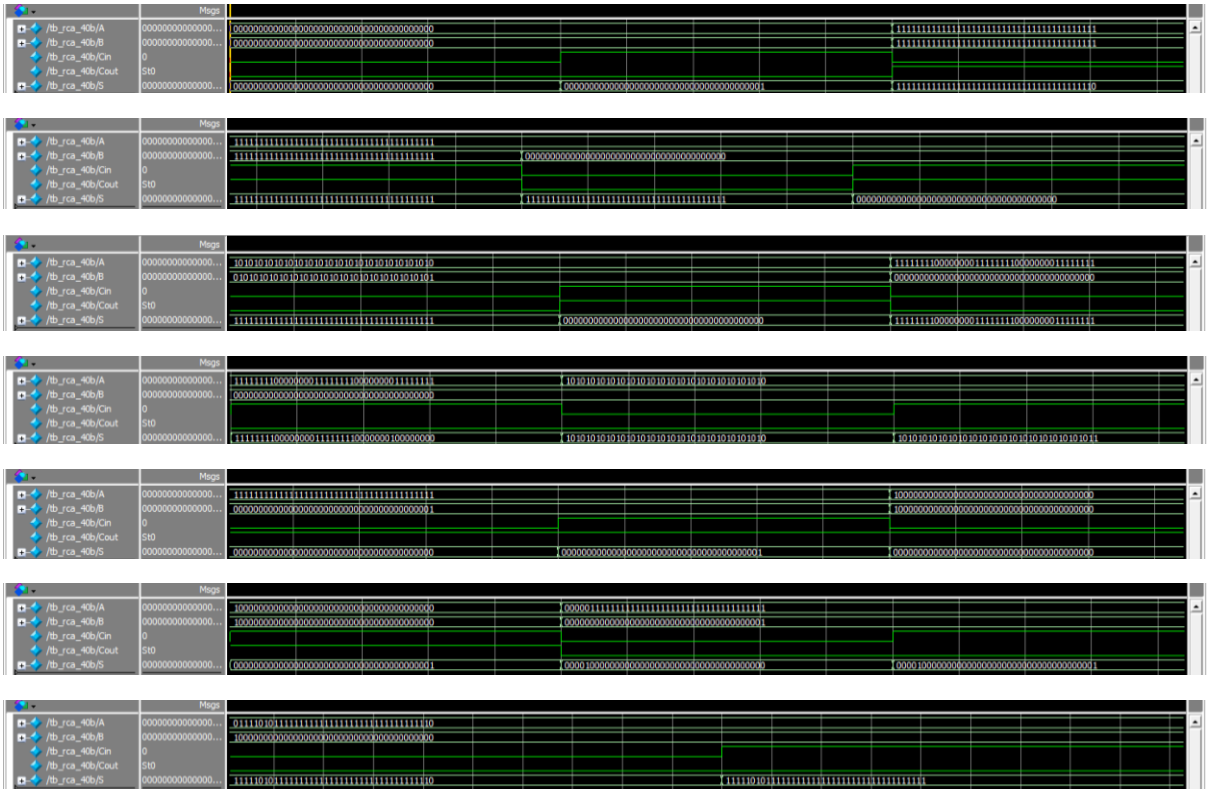


Flow Summary

<<Filter>>

Flow Status	Successful - Wed Jun 07 19:28:03 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	rca_40b
Top-level Entity Name	rca_40b
Family	Cyclone V
Device	5CGXFC7C7F23C8
Timing Models	Final
Logic utilization (in ALMs)	52 / 56,480 (< 1 %)
Total registers	0
Total pins	122 / 268 (46 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	0 / 156 (0 %)
Total HSSI RX PCSs	0 / 6 (0 %)
Total HSSI PMA RX Deserializers	0 / 6 (0 %)
Total HSSI TX PCSs	0 / 6 (0 %)
Total HSSI PMA TX Serializers	0 / 6 (0 %)
Total PLLs	0 / 13 (0 %)
Total DLLs	0 / 4 (0 %)

Testbench waveform



(A, B) 쌍으로 10 개의 경우를 채택했고 각 경우마다 Cin 이 0 일 때와 1 일 때도 경우가 나눠져 있기 때문에 총 20 가지 testbench 가 존재합니다.

Verification strategy & corresponding examples with explanation

1. A, B가 최소값을 가지는 한 쌍일 경우를 가정

   A=40'b0000000000000000000000000000000000000000
   B=40'b0000000000000000000000000000000000000000

   1) Cin=0

      S=40'b0000000000000000000000000000000000000000

      Cout=0

   2) Cin=1

      S=40'b0000000000000000000000000000000000000001

      Cout=0

2. A, B가 최대값을 가지는 한 쌍일 경우를 가정

   A=40'b1111111111111111111111111111111111111111
   B=40'b1111111111111111111111111111111111111111

   1) Cin=0

      S=40'b1111111111111111111111111111111111111110

      Cout=1

   2) Cin=1

      S=40'b1111111111111111111111111111111111111111

      Cout=1

3. A + B가 최대값을 가지는 경우를 가정

A=40'b1111111111111111111111111111111111111111
B=40'b0000000000000000000000000000000000000000

1) Cin=0

S=40'b1111111111111111111111111111111111111111

Cout=0

2) Cin=1

S=40'b0000000000000000000000000000000000000000

Cout=1

A=40'b1010101010101010101010101010101010101010
B=40'b0101010101010101010101010101010101010101

1) Cin=0

S=40'b1111111111111111111111111111111111111111

Cout=0

2) Cin=1

S=40'b0000000000000000000000000000000000000000

Cout=1

4. B=0일 경우를 가정

A=40'b1111111100000000111111110000000011111111
B=40'b0000000000000000000000000000000000000000

1) Cin=0

S=40’b11111111000000001111111100000000111111111

Cout=0

2) Cin=1

S=40’b11111111000000001111111100000001000000000

Cout=0

A=40'b10101010101010101010101010101010101010101010

B=40'b00000000000000000000000000000000000000000000

1) Cin=0

S=10101010101010101010101010101010101010101010

Cout=0

2) Cin=1

S=10101010101010101010101010101010101010101011

Cout=0

5. 항상 오버플로우가 발생할 경우를 가정

A=40'b11111111111111111111111111111111111111111111

B=40'b00000000000000000000000000000000000000000001

1) Cin=0

S=40'b00000000000000000000000000000000000000000000

Cout=1

2) Cin=1

S=40’b00000000000000000000000000000000000000000001

Cout=1

A=40'b1000000000000000000000000000000000000000

B=40'b1000000000000000000000000000000000000000

1) Cin=0

S=40’b0000000000000000000000000000000000000000

Cout=1

2) Cin=1

S=40’b0000000000000000000000000000000000000001

Cout=1

6. 항상 오버플로우가 발생하지 않는 경우를 가정

A=40'b0000011111111111111111111111111111111111

B=40'b0000000000000000000000000000000000000001

1) Cin=0

S=40’b0000100000000000000000000000000000000000

Cout=0

2) Cin=1

S=40’b0000100000000000000000000000000000000001

Cout=0

A=40'b0111101011111111111111111111111111111110

B=40'b1000000000000000000000000000000000000000

1) Cin=0

   S=40'b1111101011111111111111111111111111111110

   Cout=0

2) Cin=1

   S=40'b1111101011111111111111111111111111111111

   Cout=0

A, B가 최소값을 가지는 한 쌍이거나 최대값을 가지는 한 쌍일 경우는 특수한 경우이므로 testbench에 포함했습니다. A, B가 최소값을 가질 때는 출력 S가 Cin과 동일하며 Cout=0임을 확인했고, A, B가 최대값을 가질 때는 Cin=1일 때 출력 S가 최대값을, Cin=0일 때 출력 S가 최대값-1을 가지며 Cout=1임을 확인했습니다.

A + B가 최대값을 가지는 경우가 저의 testbench에서 가장 중요한 경우라고 생각합니다. A+B가 최대값을 가진다는 것은 40비트가 전부 1이라는 뜻이고 이는 Cin의 값에 따라 A+B가 최대값을 가질지 오버플로우로 인해 최소값을 가질지가 결정된다는 뜻이기 때문입니다. waveform을 분석한 결과, Cin=0일 때, 출력 S가 최대값, Cout=0임을 확인했고, Cin=1일 때, 출력 S가 최소값, Cout=1임을 확인했습니다.

B=0인 경우 A+B=A입니다. Cin=0이면 S=A, Cout=0이고, Cin=1이면 S=A+1인데, A가 최대값일 경우 오버플로우가 발생하여 S=0, Cout=1입니다.

항상 오버플로우가 발생하는 경우 Cin은 오로지 출력 S의 값 1 차이의 여부만을 결정합니다. 항상 Cout=1입니다.

항상 오버플로우가 발생하지 않는 경우 역시 Cin이 출력 S의 값 1 차이의 여부만을 결정하며, 항상 Cout=0입니다.

저의 RCA에서 나올 수 있는 경우들을 골라 각각 3~4개의 testbench를 추가해, 저의 testbench들이 충분히 저의 RCA code를 잘 증명할 수 있다고 생각합니다.