積體電路設計 hw4　　　　姓名：鐘民憲

學號：b06901017

<u>(a)Simulation</u>

Minimum half-cycle time: 3.337

<u>(b)Circuit diagram</u>

Part1: Squaring

(1)Version1: Using method introduced in class

1. Partial product generator

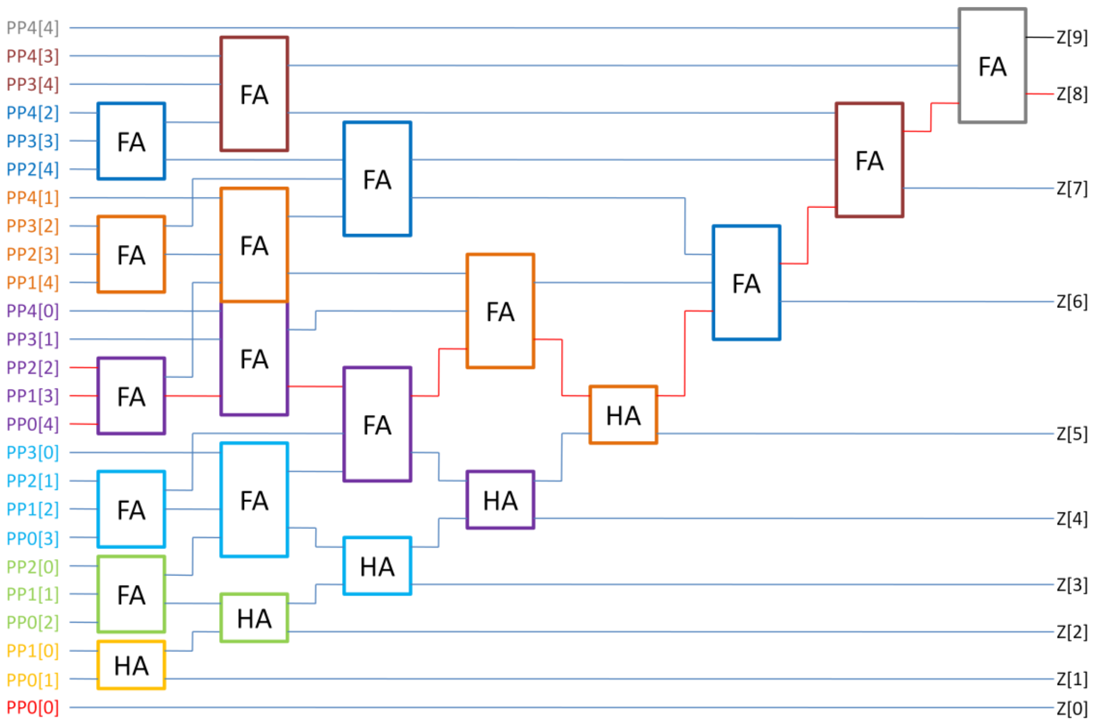
(All partial products are generated by AN2 gates)

(No single critical path)

				A[4]	A[3]	A[2]	A[1]	A[0]
				X A[4]	A[3]	A[2]	A[1]	A[0]
				PP0[4]	PP0[3]	PP0[2]	PP0[1]	PP0[0]
			PP1[4]	PP1[3]	PP1[2]	PP1[1]	PP1[0]	
		PP2[4]	PP2[3]	PP2[2]	PP2[1]	PP2[0]		
	PP3[4]	PP3[3]	PP3[2]	PP3[1]	PP3[0]			
PP4[4]	PP4[3]	PP4[2]	PP4[1]	PP4[0]				

2. Wallace tree structure

(Critical path is labeled in red)



(2)Version2: Directly apply K-map

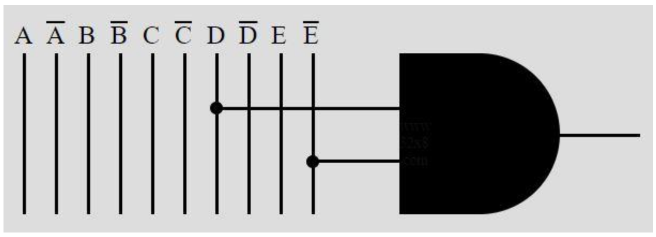
(Notation: Z[9:0]=X[5:0]^2 where A=X[4], B=X[3], C=X[2], D=X[1], E=X[0])

(Critical path appears in Z[5] and Z[6] because the OR gate has to be separated)

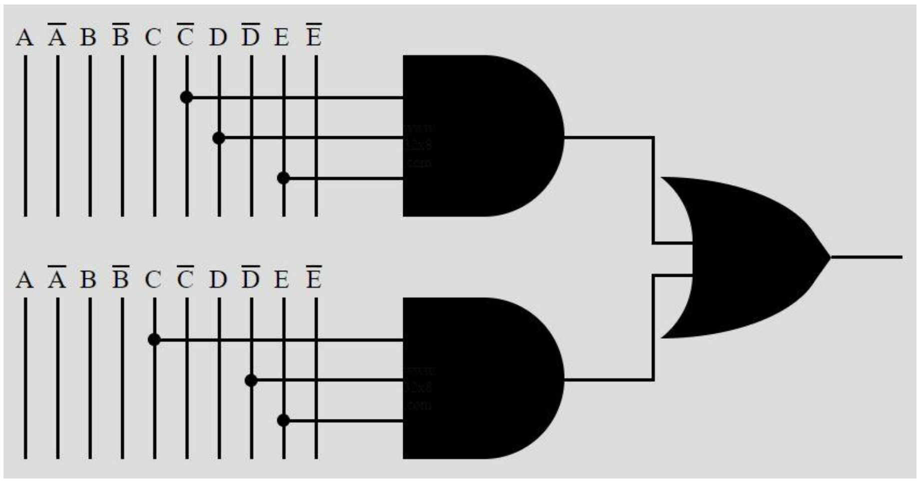
$Z[0] = E$

$Z[1] = 0$

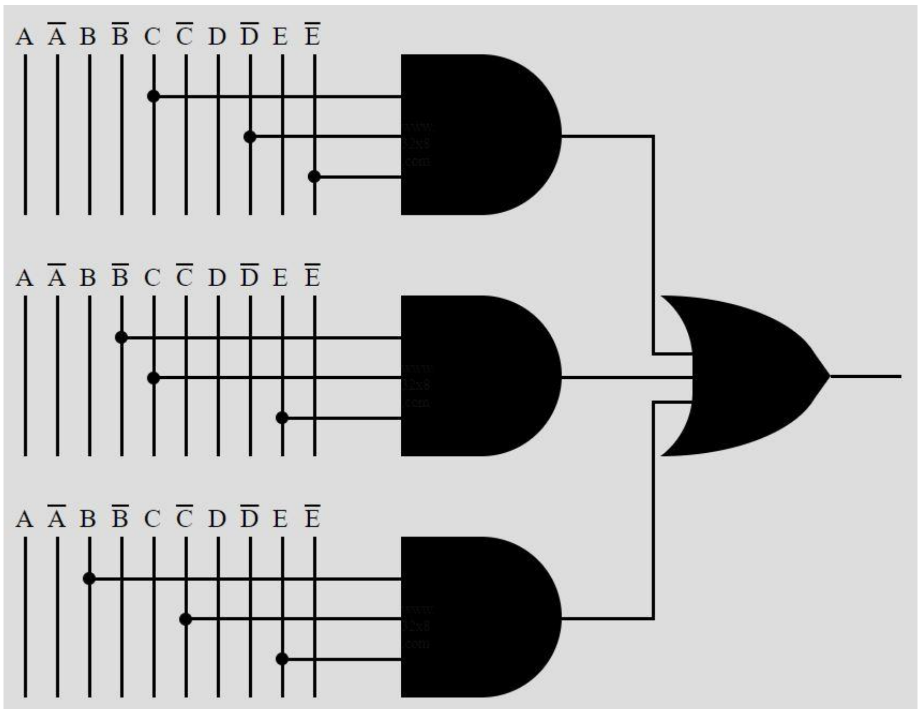
$Z[2] = D\overline{E}$



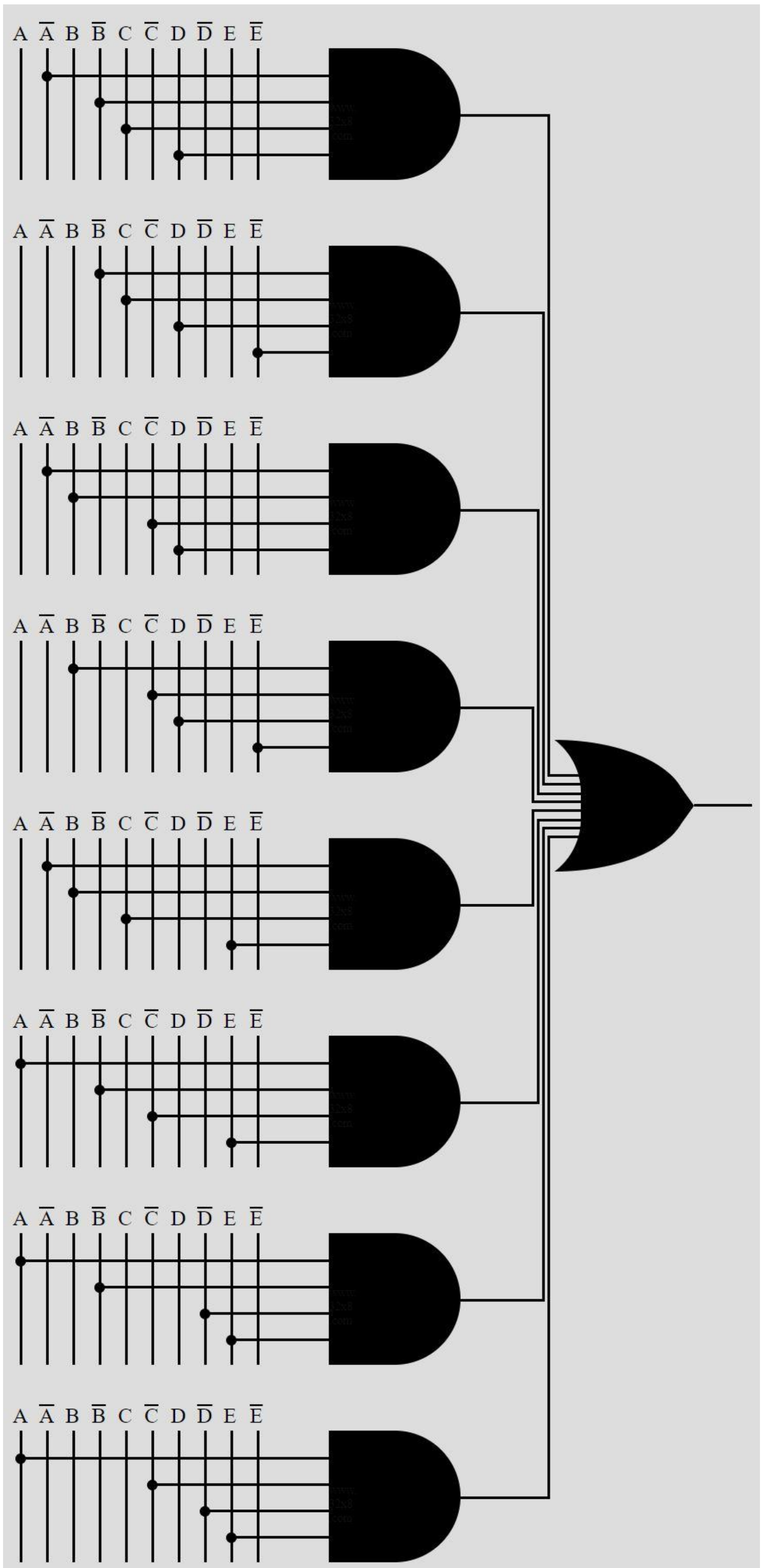
$Z[3] = \overline{C}DE + C\overline{D}E$



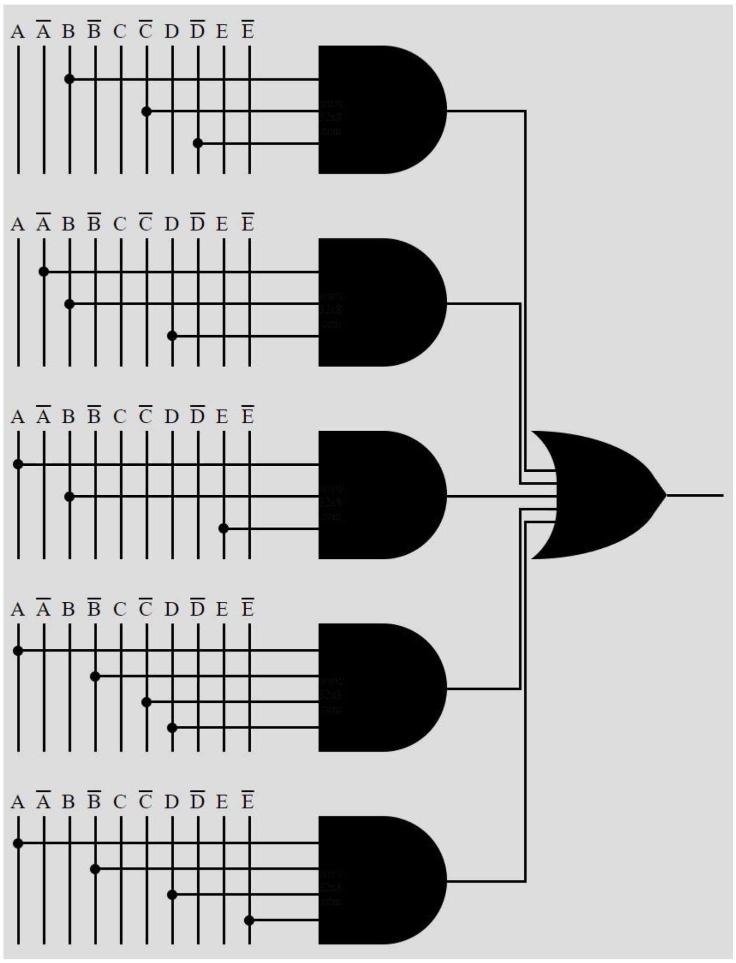
$Z[4] = C\overline{D}\overline{E} + \overline{B}CE + B\overline{C}E$



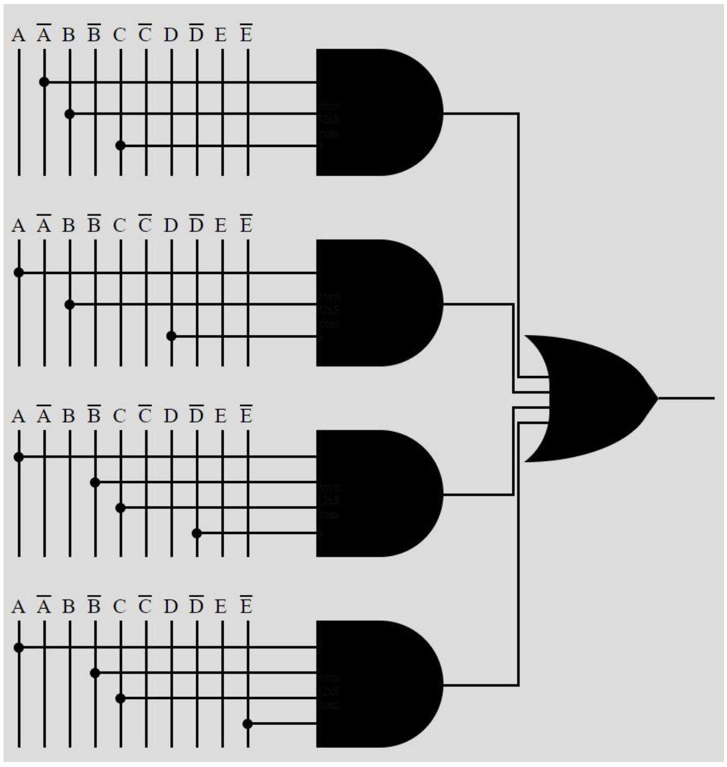
$Z[5] = \overline{A}\overline{B}\overline{C}D + \overline{B}CD\overline{E} + \overline{A}B\overline{C}D + B\overline{C}D\overline{E} + \overline{A}BCE + A\overline{B}\overline{C}E + A\overline{B}\overline{D}E + A\overline{C}\overline{D}E$



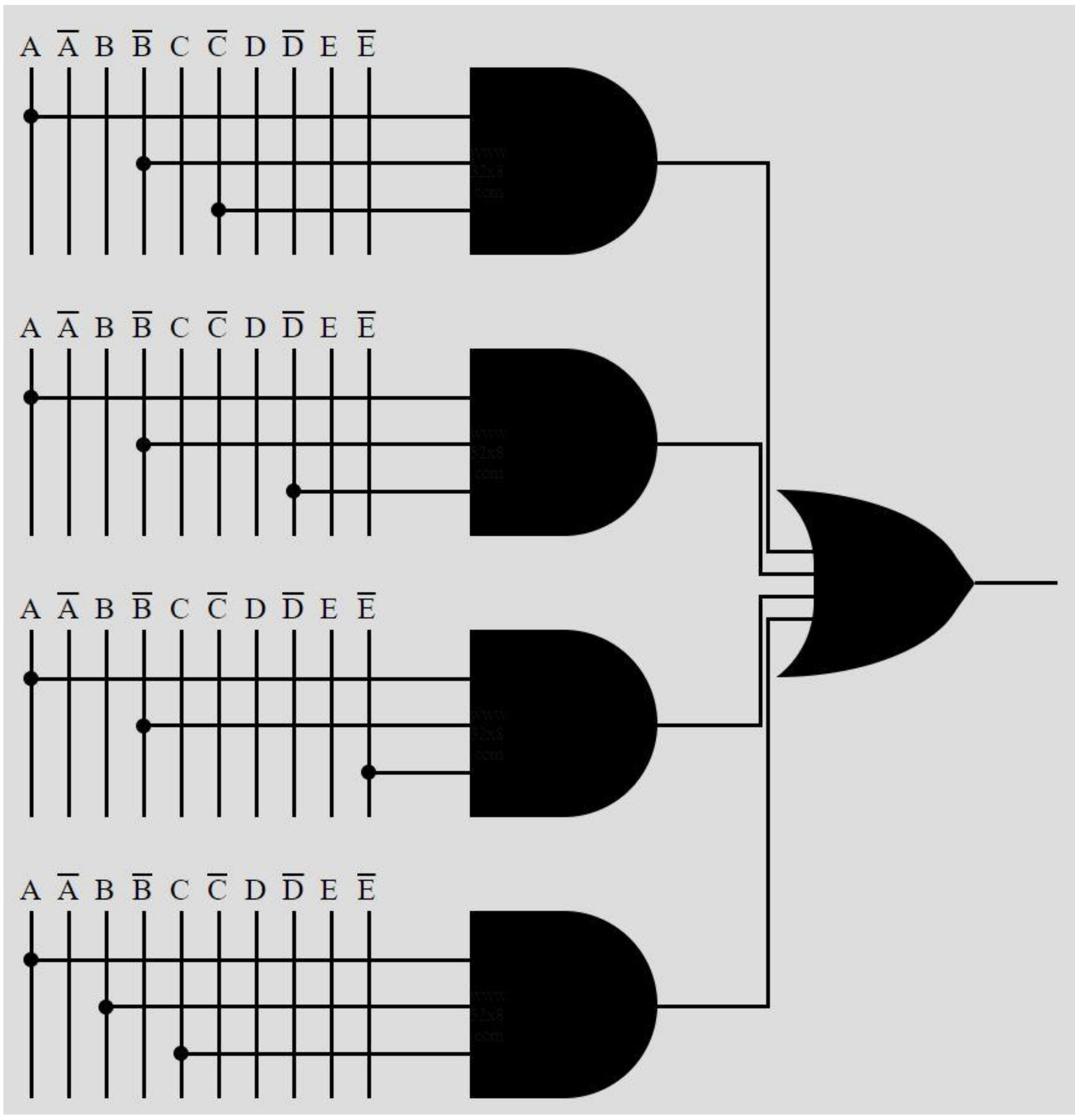
$Z[6] = B\overline{C}\overline{D} + \overline{A}BD + ABE + A\overline{B}\overline{C}D + A\overline{B}D\overline{E}$



$Z[7] = \overline{A}BC + ABD + A\overline{B}C\overline{D} + A\overline{B}C\overline{E}$

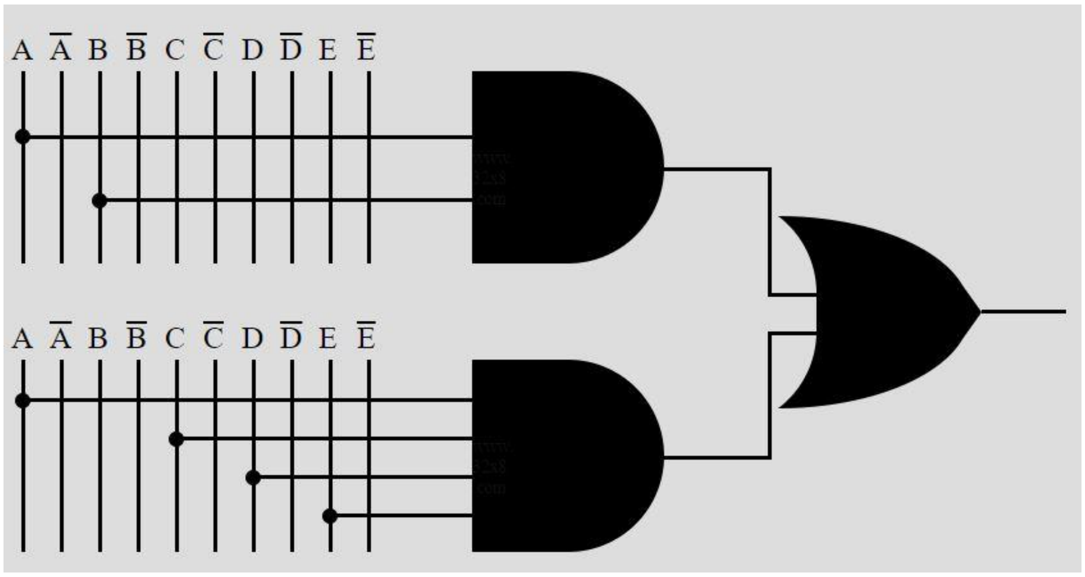


$Z[8] = A\overline{B}\overline{C} + A\overline{B}\overline{D} + A\overline{B}\overline{E} + ABC$



A $\overline{A}$ B $\overline{B}$ C $\overline{C}$ D $\overline{D}$ E $\overline{E}$
A $\overline{A}$ B $\overline{B}$ C $\overline{C}$ D $\overline{D}$ E $\overline{E}$
A $\overline{A}$ B $\overline{B}$ C $\overline{C}$ D $\overline{D}$ E $\overline{E}$
A $\overline{A}$ B $\overline{B}$ C $\overline{C}$ D $\overline{D}$ E $\overline{E}$

$Z[9] = AB + ACDE$

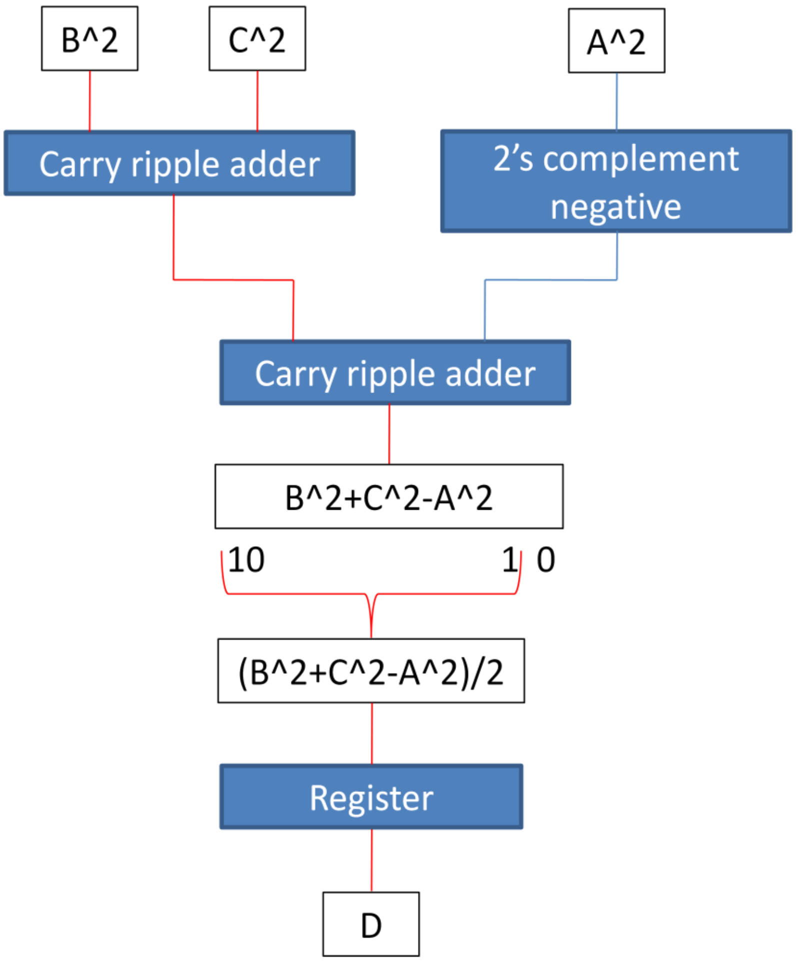


A $\overline{A}$ B $\overline{B}$ C $\overline{C}$ D $\overline{D}$ E $\overline{E}$
A $\overline{A}$ B $\overline{B}$ C $\overline{C}$ D $\overline{D}$ E $\overline{E}$

Part2: Add and subtract

(Critical path is labeled as red line)

How I do subtraction: First, inverse every bit of A^2(9-bits). Second, add a bit 1 in front of it called A^2_c. Third, add 1 to A^2_c to get A^2_n. Finally, directly add B^2+C^2 and A^2_n.



(c)Discussion

(1)My first design is simply based on the slide in class. I utilize the Wallace tree multiplier to implement the multiplication part and the carry ripple adder to implement the addition and subtraction.

After measurement and several practices, I found that the best result came out when there is no pipelining. The result is shown in the figure below.

Method1: No pipelining.

Method2: Pipeline between stage2 and stage3.

Method3: Pipeline inside stage2.

Method	Full cycle time	Area	A*T
1	7.67ns	2823	21733351.06
2	7.00ns	3660	25748100.00
3	7.00ns	3903	27457605.00

My measurement shows that stage1 takes 0.113s, stage2 takes 2.01s, stage3 takes 1.2s, and the final register layer takes 0.511s, so the best pipeline location should be in the stage2. However, in the practice, method3 performs the worst because the reduction in time doesn’t even out the increment in the area.

(2)I asked my friends how they implement squaring quickly and I got the answer: K-map aka brute force. Although I don’t think this is a way that the professor wants, I cannot deny that this is indeed the most efficient approach. The figure below shows the performance.

Method	Full cycle time	Area	A*T
K-map	6.67ns	2043	13689521.93

The result is improved gigantically. However, due to time limitation and previous experience, I didn’t try to do pipelining.