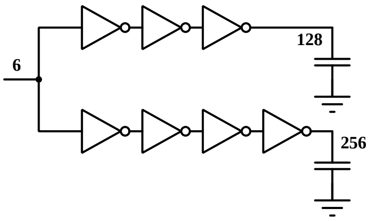
INDIAN INSTITUTE OF TECHNOLOGY BOMBAY
ELECTRICAL ENGINEERING DEPARTMENT

Solution to Class Test - 2

Quantitative answers should be accurate to 0.1%.
ALL intermediate values should be reported for iterative solutions.

Q–1 We want to design a fork using 3 inverters in one branch and 4 in the other. The total input capacitance presented by the two branches to the previous driver should be **6** times the input capacitance of the template inverter.
Assume $\mathbf{p_{inv}} = \mathbf{2.4}, \gamma = \mathbf{2.8}$.
The final load on the branch with 3 inverters is equivalent to **128** template inverters, while that on the 4 inverter branch is equivalent to **256** template inverters.



Find the input capacitances of first inverters of the two branches such that the optimum delay in the two branches is equal. Find the total delay in the two branches and verify that these are equal.

Soln. 1) Let the input capacitance of the first inverter in the 3 inverter branch be $6r$, and the input capacitance of the first inverter in the 4 inverter branch be $6(1-r)$. This ensures that the total capacitive load presented to the previous stage is 6.

$$\begin{aligned}
F_1 &= G\ B\ H = 1 \times 1 \times \frac{128}{6r} \\
F_2 &= G\ B\ H = 1 \times 1 \times \frac{256}{6(1-r)} \\
\text{Then } \hat{f}_1 &= \left(\frac{128}{6r}\right)^{1/3} = 2.773445\ r^{-1/3} \\
\text{and } \hat{f}_2 &= \left(\frac{256}{6(1-r)}\right)^{1/4} = 2.555772(1-r)^{-1/4}
\end{aligned}$$

Delay through the 3 inverter branch is $3\hat{f}_1 + 3p_{inv}$, while that through the 4 inverter branch is $4\hat{f}_2 + 4p_{inv}$. Equating the two, we get

$$4 \times 2.555772(1-r)^{-1/4} + 4p_{inv} - 3 \times 2.773445\ r^{-1/3} - 3p_{inv} = 0$$

which gives us the non-linear equation

$$f(r) \equiv 10.22309(1-r)^{-1/4} + 2.4 - 8.320335r^{-1/3} = 0$$

This can be solved using Newton Raphson iterations. Taking the derivative with respect to r, we get

$$f'(r) = \frac{10.22309}{4}(1-r)^{-5/4} + \frac{8.320335}{4}(r)^{-4/3} = 2.555772(1-r)^{-5/4} + 2.773445r^{-4/3}$$

Let us start with the initial guess of $r = 0.5$ (equal drive to the two branches).
successive iterations give:

g	f(g)	f'(g)	g_{next}
0.5	4.074405	13.067333	0.1881991
0.1881991	-1.348842	29.03231	0.2346591
0.2346591	-0.1595177	22.73209	0.2416764
0.2416764	-2.475290×10^{-3}	22.03521	0.2417887
0.2417887	-6.035093×10^{-7}	22.02447	0.2417888
0.2417888	-4.08×10^{-14}	22.02447	0.2417888

So the solution converges to $r = 0.2417888$.

Therefore input capacitances are:
$6r = 6 \times 0.2417888 = 1.450733 \approx 1.45$ for the branch with 3 inverters, and
$6(1-r) = 4.549267 \approx 4.55$ for the branch with 4 inverters.

$$\begin{aligned}
\hat{f}_1 &= \left(\frac{128}{6r}\right)^{1/3} = 4.451854 \\
\hat{f}_2 &= \left(\frac{256}{6(1-r)}\right)^{1/4} = 2.738890 \\
\text{Delay}_1 &= 3\hat{f}_1 + 3p_{inv} = 3 \times (4.451854 + 2.4) = 20.55556 \\
\text{Delay}_2 &= 4\hat{f}_2 + 4p_{inv} = 4 \times (2.738890 + 2.4) = 20.55556
\end{aligned}$$

The two delays are identical, as desired. – **[Q1: 3 marks]**

Q–2 Illustrate multiplication of two 6 bit unsigned numbers 45(decimal) and 26(decimal) where partial products are generated using (modified) Booth algorithm.
The partial products should be reported as binary numbers of appropriate width.
Show by adding the partial products sequentially that binary addition of partial products gives the correct result.
(All partial products must be sign extended to appropriate width).

Soln. 2) Multiplicand and multiplier can be accommodated in 6 bits each. So the product should be 12 bit wide.
$A = 45_{\text{decimal}} = 101101, \quad B = 26_{\text{decimal}} = 011010.$
We can pre-compute the possible 12 bit partial product terms as:
$2A = 000001011010, -A = 111111010011$ (sign extended to 12 bits),
and $-2A = 111110100110$ (sign extended to 12 bits).
Now we scan the multiplier from the right 2 bits by 2bits with an overlapping most significant bit from the previous group.

Multiplier bits	Partial Product	Binary Partial Product
10:0	-2A	111110100110
10:1	-A	1111010011xx
	sum	111011110010
01:1	2A	01011010xxxx
	sum	010010010010

So the result is $0x492 = 4 \times 256 + 9 \times 16 + 2 = 1024 + 144 + 2 = 1170$.
This agrees with $45 \times 26 = 1170$. – **[Q2: 3 marks]**

Q–3 A Multiply and Accumulate circuit which multiplies two 6 bit numbers and adds an 8 bit number to the product is to be designed.

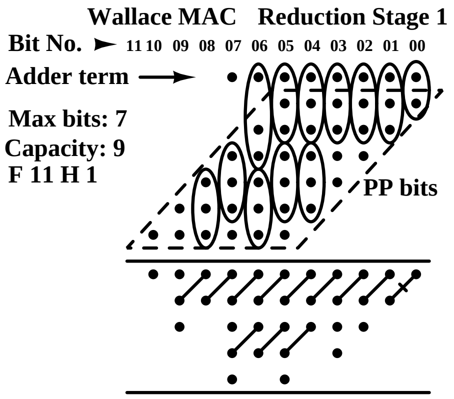
a) Show the wire reduction scheme with a dot diagram using the Wallace algorithm which does not generate redundant bits. How many half adders and full adders are used in this case? What is the width of the final carry propagating adder required after wire reduction?

Soln. 3-a) The 6x6 multiplier results in 6 partial products to be added. The 8 bit operand to be added to the product results in 7 rows which need to be added. The capacities of reduction stages counting from the final stage are: $2, 3, 4, 6, 9 \cdots$. So we can begin with a capacity of 9, which is the smallest capacity value which will accommodate 7 bits in a column.

At every reduction stage, we follow the following rules:

1. Wherever we find bunches of 3 wires at the same weight, we feed these to a full adder.
2. If two wires are left over:
   (a) If there is one wire or less in all the columns to the right, the two wires are reduced using a half adder.
   (b) If the capacity of the next stage will not be exceeded by passing these through, both wires are passed through to the next stage
   (c) If the capacity will be exceeded, the two wires are fed to a half adder.
3. If one wire is left over, it is passed through to the next stage.

First Reduction Stage: At the start, the number of wires at bit positions 10 to 0 are:
1, 2, 3, 5, 6, 7, 6, 5, 4, 3 and 2. Capacity of the next level will be 6.

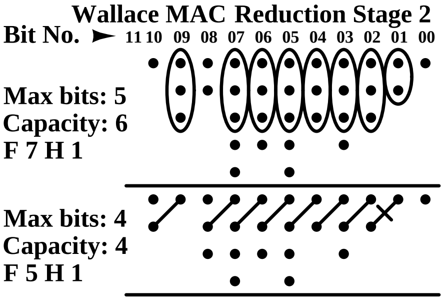

Wallace MAC Reduction Stage 1
Bit No. ► 11 10 09 08 07 06 05 04 03 02 01 00
Adder term
Max bits: 7
Capacity: 9
F 11 H 1
PP bits

Full adders are placed wherever there are bunches of 3 wires at the same weight. (1 each at bits 1, 2, 3, 7 and 8; 2 each at bits 4, 5 and 6)
2 wires at bit 0 are reduced by a half adder because there are no wires to the right of this column. Other occurrences of 2 left over wires (at bits 3 and 9) are fed through because the next layer, which has a capacity of 6, can accommodate these.
All left over single wires (at bits 0, 2, 5 and 10) are fed through to next stage.

Bit No.	11	10	09	08	07	06	05	04	03	02	01	00	Total
In		1	2	3	5	6	7	6	5	4	3	2	Adders
FA				1	1	2	2	2	1	1	1		11
HA												1	1
Out		1	3	2	5	4	5	3	4	3	2	1	

11 full adders and 1 half adder will be used in this stage as shown.

Reduction Stage 2: Incoming wires at bit positions 10 to 0 are:
1, 3, 2, 5, 4, 5, 3, 4, 3, 2, 1. Capacity of next stage is 4.

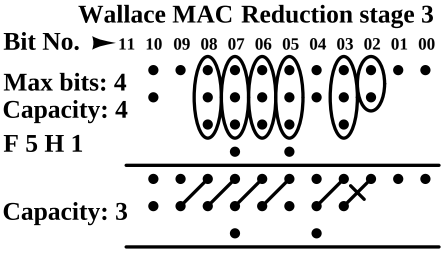

Wallace MAC Reduction Stage 2
Bit No. ➤ 11 10 09 08 07 06 05 04 03 02 01 00
Max bits: 5
Capacity: 6
F 7 H 1
Max bits: 4
Capacity: 4
F 5 H 1

Full adders are placed at bits 2, 3, 4, 5, 6, 7 and 9. A half adder is used at bit 1 since all columns to the right of it have 1 wire or less. 2 left over wires at bits 5 and 7 are fed through because these can be accommodated within the next layer's capacity of 4. Left over Single wires at bits 0, 3 and 6 are fed through.

Bit No.	11	10	09	08	07	06	05	04	03	02	01	00	Total
In		1	3	2	5	4	5	3	4	3	2	1	
FA			1		1	1	1	1	1	1			7
HA											1		1
Out		2	1	3	4	3	4	2	3	2	1	1	

7 full adders and 1 half adder are used by this stage.

Reduction Stage 3: Incoming wires at bit positions 10 to 0 are:
2, 1, 3, 4, 3, 4, 2, 3, 2, 1 and 1.

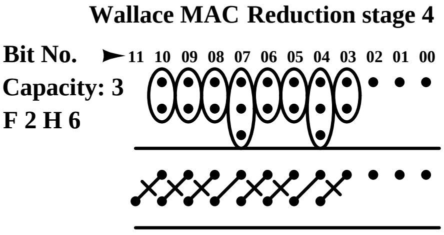

Wallace MAC Reduction stage 3
Bit No. ➤ 11 10 09 08 07 06 05 04 03 02 01 00
Max bits: 4
Capacity: 4
F 5 H 1
Capacity: 3

Full adders are placed at bits 3, 5, 6, 7 and 8. Two wires at bit 2 are reduced by a half adder since all columns to the right of it have a single wire. Two wires at bits 4 and 10 are passed through since these can be accommodated by the capacity of next stage (=3). Single left over wires at bits 5 and 7 are passed through.

Bit No.	11	10	09	08	07	06	05	04	03	02	01	00	Total
In		2	1	3	4	3	4	2	3	2	1	1	
FA				1	1	1	1		1				5
HA										1			1
Out		2	2	2	3	2	2	3	2	1	1	1	

This stage uses 5 Full adders and a half adder.

Reduction Stage 4 Incoming wires at bit positions 10 to 0 are:
2, 2, 2, 3, 2, 2, 3, 2, 1, 1 and 1. This is the last stage and needs to reduce wires to two or less at all bit positions.


Wallace MAC Reduction stage 4
Bit No. ➤ 11 10 09 08 07 06 05 04 03 02 01 00
Capacity: 3
F 2 H 6

Full adders are placed at bits 4 and 7. Two wires at bit 3 are reduced by a half adder since all columns to the right of it have one wire or less. Two wires bits 5, 6, 8, 9, 10 are all reduced by half adders to keep the wires at the output to 2 or less.

	at Bit No.												Total
	11	10	09	08	07	06	05	04	03	02	01	00	Adders
In		2	2	2	3	2	2	3	2	1	1	1	
FA					1			1					2
HA		1	1	1		1	1		1				6
Out	1	2	2	2	2	2	2	2	1	1	1	1	

This stage uses 2 Full adders and 6 Half adders

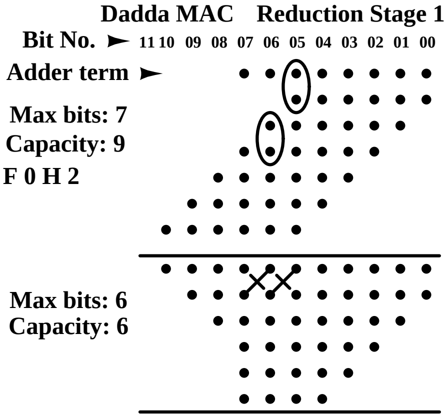
Thus this scheme uses a total of $11+7+5+2 = 25$ Full adders and $1+1+1+6 = 9$ Half adders. 4 right hand columns have a single wire. Therefore, we need to use an 8 bit final adder. – [**2**]

b) Show the wire reduction scheme with a dot diagram using the Dadda algorithm. How many half adders and full adders are used in this case? What is the width of the final carry propagating adder required after wire reduction?

Soln. 3-b) As described in the case of Wallace multiplier above, we begin with the maximum number of wires at any weight being 7, so the first reduction layer will have a capacity of 9 which is then reduced to 6, 4, 3 and 2 in subsequent layers.

In Dadda multipliers, we use the *smallest number* of *smallest* adders which will keep the number of wires at any weight within the capacity of the next layer.
The following table shows the adders which should be used in various stages to keep the max. number of wires within the capacity of each stage. below.

First Reduction Stage: At the start, the number of wires at bit positions 10 to 0 are: 1, 2, 3, 5, 6, 7, 6, 5, 4, 3 and 2. Capacity of the next level is 6.

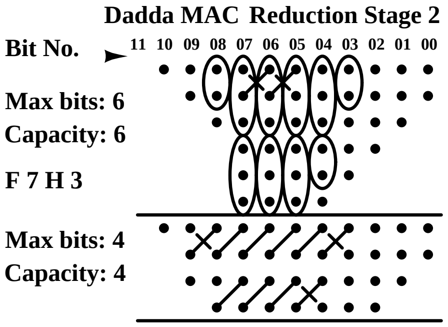

Dadda MAC Reduction Stage 1
Bit No. ► 11 10 09 08 07 06 05 04 03 02 01 00
Adder term ►
Max bits: 7
Capacity: 9
F 0 H 2
Max bits: 6
Capacity: 6

All columns on the right which have 6 wires or less (bits 0 to 4) are just passed through. Bit 5 has 7 wires, so the count is reduced by 1 using a half adder. Its carry will go to bit 6, taking its wire count to 7. Therefore this column should also reduce the wire count by 1 using a half adder. All other columns will now have six wires or less, so no other action is required.

Bit	11	10	09	08	07	06	05	04	03	02	01	00	Total
In		1	2	3	5	6	7	6	5	4	3	2	Adders
FA													0
HA						1	1						2
Out		1	2	3	6	6	6	6	5	4	3	2	

So this stage uses 2 half adders and no full adders.

Reduction stage 2: In coming wires at bit positions 10 to 0 are:
1, 2, 3, 6, 6, 6, 6, 5, 4, 3 and 2. Capacity of the next stage is 4.
Bits 0, 1 and 2 have 4 wires or less, so these are passed through.

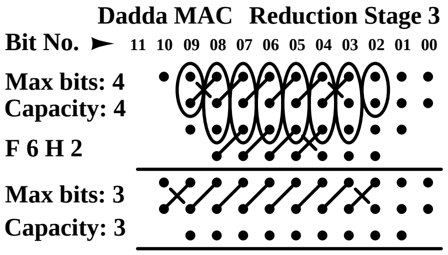


Bit 3 has 5 wires, so we reduce the count by 1 using a half adder. Its carry goes to Bit 4, which will then have 7 wires. It is reduced by 3 using a full and a half adder. The two carry outputs of these make the wire count at bit 5 reach 8. This is reduced to 4 by using two full adders. The same is true of bits 6 and 7. Two carries from bit 7 make the wire count at bit 8 reach 5, which we reduce by 1 using a half adder. Other columns can just be passed through.

Bit	11	10	09	08	07	06	05	04	03	02	01	00	Total
In		1	2	3	6	6	6	6	5	4	3	2	Adders
FA					2	2	2	1					7
HA				1				1	1				3
Out		1	3	4	4	4	4	4	4	4	3	2	

So this stage uses 7 full adders (2 each at bits 5, 6 and 7, 1 at bit 4) and 3 half adders (at bits 3, 4 and 8).

Reduction stage 3: In coming wires at bit positions 10 to 0 are: 1, 3, 4, 4, 4, 4, 4, 4, 4, 3 and 2. Capacity of the next stage is 3 so wires of bits 0 and 1 can be passed through.

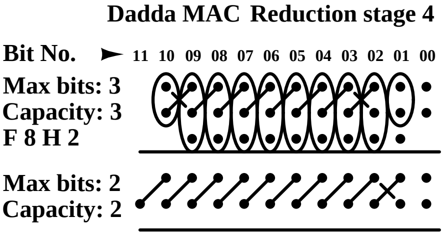


Bit 2 has 4 wires, which are reduced to 3 using a half adder. The carry from this makes the wire count at bit 3 to 5, which is reduced to 3 using a full adder. This is so for bits 4,5,6,7 and 8 as well. The carry from the full adder at bit 8 takes the wire count at bit 9 to 4, which is reduced to 3 using a half adder. Column 10 can just be passed through.

Bit	11	10	09	08	07	06	05	04	03	02	01	00	Total
In		1	3	4	4	4	4	4	4	4	3	2	Adders
FA				1	1	1	1	1	1				6
HA			1							1			2
Out		2	3	3	3	3	3	3	3	3	3	2	

This stage uses 6 full adders (at bits 3, 4, 5, 6, 7 and 8) and 2 half adders (at bits 2 and 9).

Reduction stage 4: In coming wires at bit positions 10 to 0 are: 2, 3, 3, 3, 3, 3, 3, 3, 3, 3 and 2. Capacity of the next stage is 2, so wires of bit 0 can be passed through.



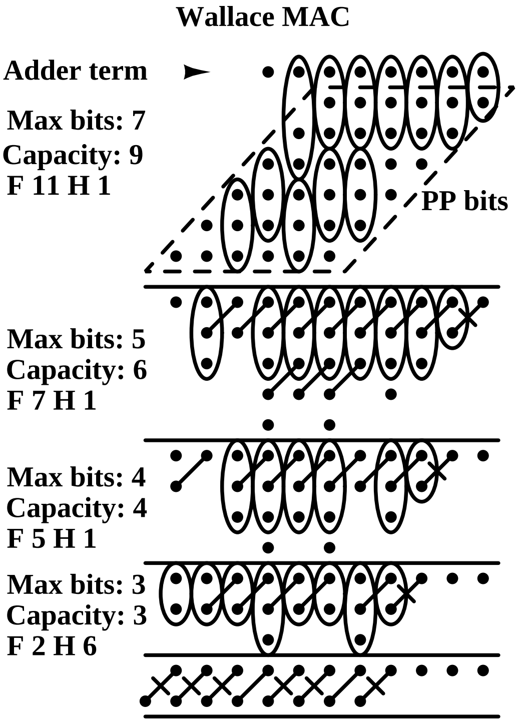
Bit 1 has 3 wires which are reduced to 2 using a half adder. The carry from this will take the wire count to 4 at bit 2, which is reduced to 2 using a full adder. This is so for bits 3 to 9 as well. Carry from bit 9 will take the wire count at bit 10 to 3, which can be reduced to 2 using a half adder.

Bit	11	10	09	08	07	06	05	04	03	02	01	00	Total
In		2	3	3	3	3	3	3	3	3	3	2	Adders
FA			1	1	1	1	1	1	1	1			8
HA		1									1		2
Out	1	2	2	2	2	2	2	2	2	2	2	2	

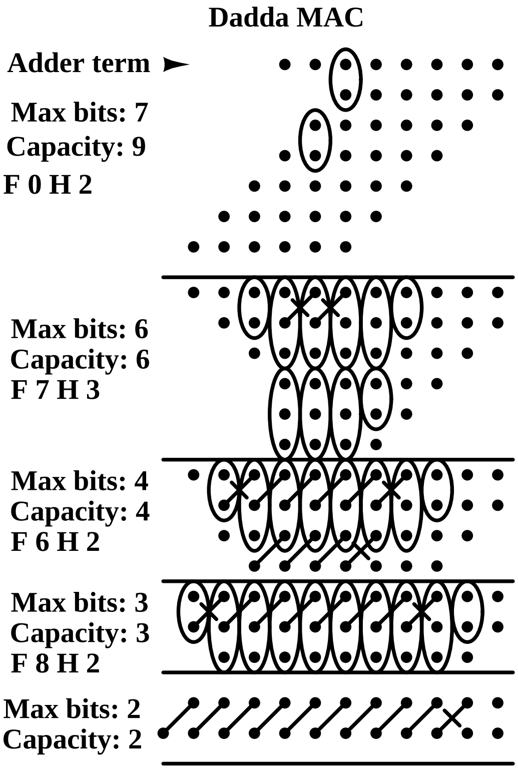
So this stage uses 8 full adders (at bits 2 to 9) and 2 half adders (at bits 1 and 10). A 12 bit final adder will be required. Notice that we could have reduced the 2 wires at bit 0 using a half adder, replacing the half adder at bit 1 by a full adder. Rest of the arrangement remains the same. This option will use 9 full adders (at bits 1 to 9) and two half adders (at bits 0 and 10), with an 11 bit final adder.

– [**2**]

Dot diagrams for the complete wire reduction schemes in the two cases are shown below:



Full adders: $11 + 7 + 5 + 2 = 25$
Half adders: $1 + 1 + 1 + 6 = 9$



Full adders: $0 + 7 + 6 + 8 = 21$
Half adders: $2 + 3 + 2 + 2 = 9$