EC 39003 Digital Circuits Lab

04/11/2020

Experiment on Implementing (4X4)-bit Serial-Parallel Multiplier

$a_3\, a_2\, a_1\, a_0$ – multiplicand

$b_3\, b_2\, b_1\, b_0$ – multiplier

PP_0:				a_3b_0	a_2b_0	a_1b_0	a_0b_0	
PP_1:			a_3b_1	a_2b_1	a_1b_1	a_0b_1		×
PP_2:		a_3b_2	a_2b_2	a_1b_2	a_0b_2		×	×
PP_3:	a_3b_2	a_2b_3	a_1b_3	a_0b_3		×	×	×
	R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0

Let $a_3a_2a_1a_0 \times b_0 = PP_0$

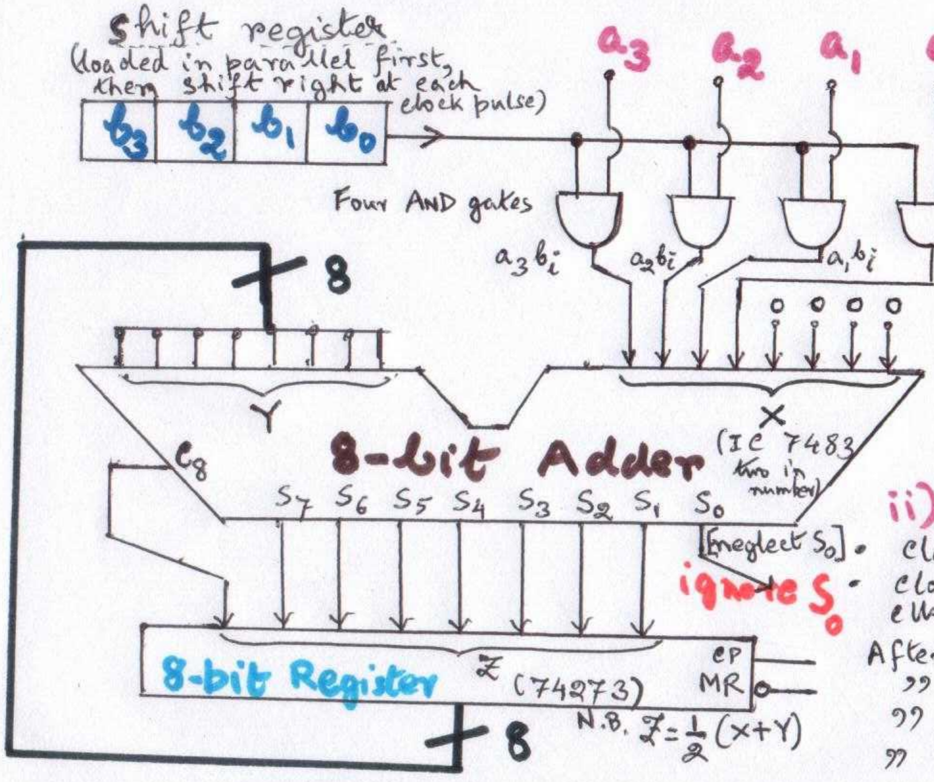
$a_3\,a_2\,a_1\,a_0 \times b_1 = PP_1$

$a_3\,a_2\,a_1\,a_0 \times b_2 = PP_2$

$a_3\,a_2\,a_1\,a_0 \times b_3 = PP_3$

Summing up the partial products PP_i, $i = 0, 1, 2, 3$ with proper weights, the final product is obtained as

$$FP = 8PP_3 + 4PP_2 + 2PP_1 + PP_0$$


Shift register
(loaded in parallel first, then shift right at each clock pulse)
b_3 b_2 b_1 b_0
a_3 a_2 a_1 a_0
Four AND gates
a_3b_i a_2b_i a_1b_i a_0b_i
8
Y
X
8-bit Adder
(IC 7483 two in number)
c_8
S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0
[neglect S_0]
ignore S_0
8-bit Register
Z
(74273)
CP
MR
N.B. $Z = \frac{1}{2}(X+Y)$
8

Required components

(i) IC 7495 (4-bit shift register)

(ii) IC 7483 (4-bit adder) – two are required

(iii) IC 74273 (octal +ve edge triggered DFF with active low Master Reset)

(iv) IC 7408 (four 2-input AND gates (v) logic switches to feed in 4-bit multiplicand and multiplier

(vi) logic displays (simple lamps) to read product register.

Major steps:

i) Load 4-bit multiplier $(b_3b_2b_1b_0)$ into shift register IC 7495 and multiplicand $(a_3a_2a_1a_0)$ by logic switches

ii) feed manual clk pulses to IC 7495 and IC 74273

- Clear product register (74273) to all zeroes.
- Clock for 7495 $\downarrow$ while it is $\uparrow$ for 273. Invert the clk (fed to 95) by one or three inverters & feed to 273.

After 1st clk pulse : $X: 16PP_0$; $Y: 0$; $Z: 8PP_0$ $(16PP_0/2)$

” 2nd clk pulse : $X: 16PP_1$; $Y: 8PP_0$; $Z: \frac{1}{2}(16PP_1 + 8PP_0) = 8PP_1 + 4PP_0$

” 3rd clk pulse : $X: 16PP_2$; $Y: 8PP_1 + 4PP_0$; $Z: \frac{1}{2}(16PP_2 + 8PP_1 + 4PP_0) = 8PP_2 + 4PP_1 + 2PP_0$

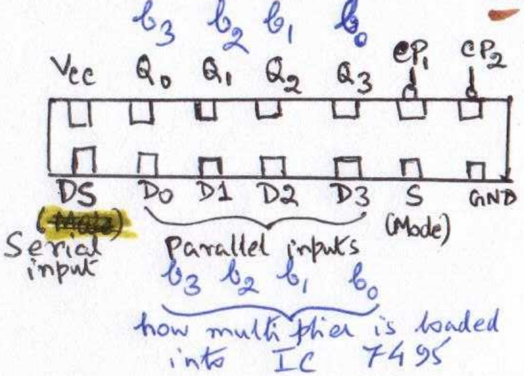
” 4th clk pulse : $X: 16PP_3$; $Y: 8PP_2 + 4PP_1 + 2PP_0$; $Z: \frac{1}{2}(16PP_3 + 8PP_2 + 4PP_1 + 2PP_0)$
$= 8PP_3 + 4PP_2 + 2PP_1 + PP_0 = FP$

EC 39003 Digital Circuits Lab

04/11/2020

i) IC 7495 4-bit shift register

with serial and parallel synchronous operating modes



S : mode control input

$D_0 - D_3$: parallel data inputs

DS : serial data input

$\overline{CP_1}$: serial clock (active low going edge) input

$\overline{CP_2}$: parallel clock (active low going edge) input

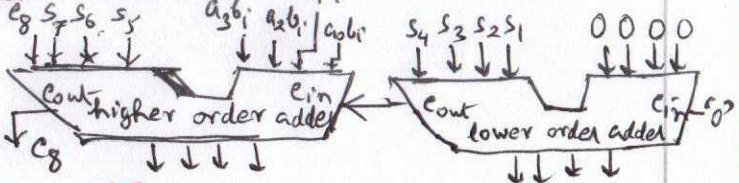
$Q_0 - Q_3$: parallel outputs

- When S is HIGH ("1"), $\overline{CP_2}$ is enabled. A hi-to-lo (⌊) transition on enabled $\overline{CP_2}$ transfers parallel data from the $P_0 - P_3$ inputs to the $Q_0 - Q_3$ outputs (parl load)
- When S is LOW ("0"), $\overline{CP_1}$ is enabled. A high-to-low (⌊) transition on enabled $\overline{CP_1}$ transfers the data from serial input (DS) to Q_0, and shifts the data from Q_0 to Q_1, Q_1 to Q_2, and Q_2 to Q_3 respectively (right shift)
- For normal operation, S should be allowed to change states when both the clock i/ps are LOW.

N.B. Keep CP₁ and CP₂ shorted as parallel load & right shift are to occur at different times.

ii) IC 7483 - 4-bit parallel adder with Cin, Cout

The two adders will be connected in cascade as shown here ⇒



ii) IC 74273 - octal positive edge triggered DFFs with active low clear ($\overline{MR}$) input:

note that the FFs in 9273 are positive edge triggered whereas, the FFs in 7495 are negative edge triggered. So, one can generate clk for 7495 ⌊ by way of logic switch, and invert this signal using one or better three inverters (to provide adequate delay) and feed the inverted clock to CP input of 74273.

iii) IC 7408 - quad 2-input NAND gates

iv) Use logic display (simple lamps)

to read the value of the contents of the product register IC 74273