

به نام خدا

سرکار جناب آقای دکتر اسدی

سارا آذرنوش 98170668

تمرین 3

درس معماری کامپیوتر

(1)

Group size =4

N = 64

Dg = 2

$2 \times (n/4) \times dg \Rightarrow \text{carry using rc}$

+

$Dg \Rightarrow pi \text{ and } gi$

+

$2 \times dg \Rightarrow \text{carry through a gp}$

$((2 \times (n/4)) + 3) \times dg = 70$

(2)

بیتی به عنوان مضروب در نظر گرفته شود: $m+2$ و عدد اگر شرطی دراینکه کدام عدد مضروب باشد موجود نباشد

نیاز است درحالت مینیموم به صورت 1 در چپترین و دیگر بیتها 0 باشند به 1 عملیات

عملیات نیاز است $m+2$ (راستترین 1 باشد که خود نیز یک عملیات نیاز دارد) به و ماکسیمم به حالت یکی درمیان 0 و 1 است

*$m * (m+2)$*

$Min \Rightarrow 10...$

$Max \Rightarrow 101...$

(3)

$98170668 \Rightarrow (70668)_{10} = (10001010000001100)_2$

به علت دشواری محاسبه گفته شد ابتدا به باینری تبدیل کرده و سپس 5 رقم جدا کنیم

$(98170668)_{10} = (101110110011111011100101100)_2$

$01100 \times 10010 = 011011000$

10010

*

01100

00000

000000

-1001000

00000000

+10010000

011011000

بیت‌های ضرب کننده را در نظر می‌گیریم و بررسی می‌کنیم:

- اولین عدد سمت چپ 0 است بنابراین یک شیفت می‌دهیم (به تعداد بیت‌ها 0 می‌گذاریم)
- رقم بعدی نیز 0 میباشد بنابراین چون 00 است طبق الگوریتم تنها شیفت می‌دهیم (یکی بیشتر از تعداد بیت عدد بالایی 0 می‌گذاریم) و عملیات محاسباتی انجام نمیشود.
- 10 داریم و طبق الگوریتم مضروب را کم میکنیم ($x = x' + 1$)
- در مرحله بعد نیز 11 داریم که مطابق بالا شیفت می‌دهیم.
- و در آخر 01 داریم و مضروب را جمع میکنیم سپس همه عبارات را جمع میکنیم.

$$-1001000 = 0110111 + 1 = 1001000$$

(4)

(5)

Multiplicand در تعاریف مختلف اینترنت جایگاه مختلفی دارد در اسلایدهای استاد دومین عدد میباشد

$$(15)_{10} = (01111)_2$$

$$(13)_{10} = (01101)_2$$

$$(-13)_{10} = (10011)_2$$

در ابتدا به عدد 1 میرسیم و بنابراین جمع عدد بالا را میگذاریم و سپس بقیه الگوریتم بوث را ادامه میدهیم:

$$\begin{array}{r}
 01101 \\
 * \\
 01111 \\
 \hline
 -10011 \\
 000000 \\
 0000000 \\
 00000000 \\
 +011010000 \\
 \hline
 11000011
 \end{array}$$

$$\begin{array}{r}
 10011 \\
 * \\
 01111 \\
 \hline
 -10011 \\
 000000 \\
 0000000 \\
 00000000 \\
 +011010000 \\
 \hline
 010111101
 \end{array}$$

$$Delay*cost = min$$

$$DELAY = (N/K)*DFA + (K-1)*DMUX$$

$$COST = N*((2K-1)/K)*FA + (K-1)*MUX$$

$$((8/k) \times 3 + (k - 1) \times 3) \times (8 \times ((2k - 1)/k) \times 3 + (k - 1) \times 6) = min$$

$$K=1$$

(7)

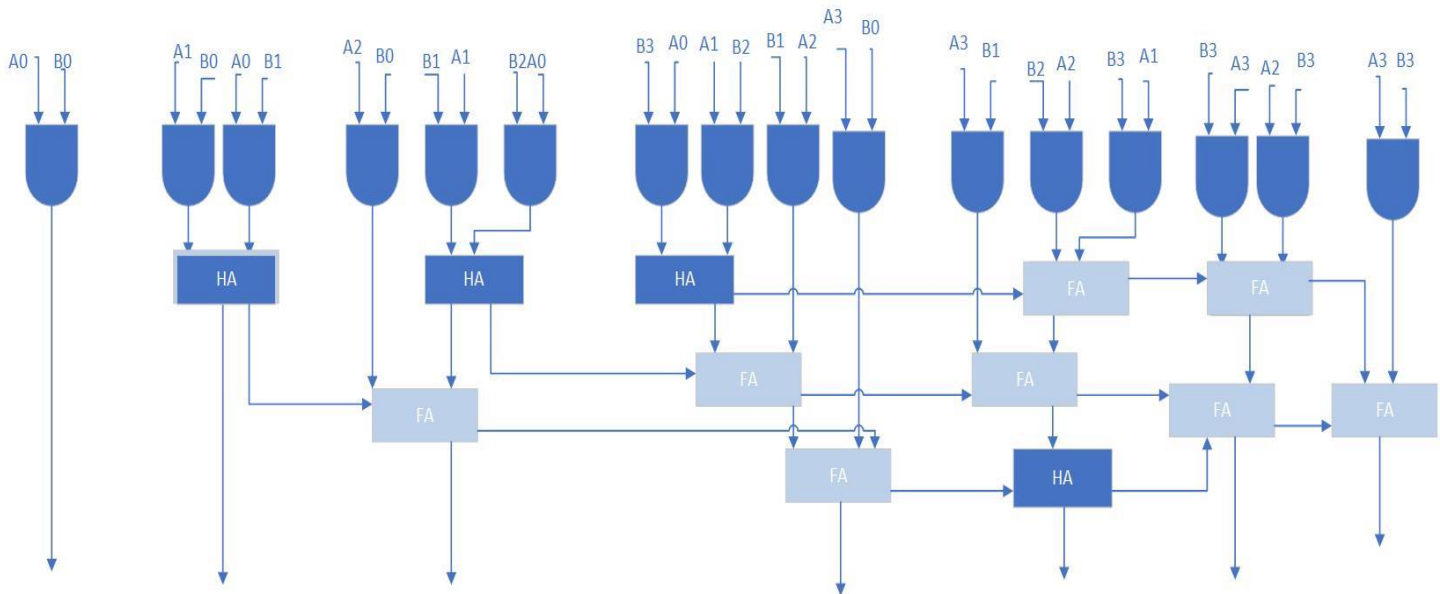
Combinational multiplexer

با توجه به الگوریتم نحوه محاسبه مدار باید به شکل زیر باشد:

الگوریتم:

مانند ضرب ساده عبارات را نوشته سپس بیت هایی که در هم ضرب شده اند با استفاده از گیت *and* بدست آورده و آن ها که زیر هم قرار دارند با هم جمع میشوند (آن هایی که از قبلی کپی ندارند با *HA* و آن هایی که دارند با *FA* محاسبه میشوند) (*N-1* بخش اول و و بخش *N-1* کپی پشین ندارند و با *HA* محاسبه میشوند و بقیه *FA*)

		A₃	A₂	A₁	A₀
		B₃	B₂	B₁	B₀
		<hr/>			
		A₃ . B₀	A₂ . B₀	A₁ . B₀	A₀ . B₀
	A₃ . B₁	A₂ . B₁	A₁ . B₁	A₀ . B₁	
A₃ . B₂	A₂ . B₂	A₁ . B₂	A₀ . B₂		
A₂ . B₃	A₁ . B₃	A₀ . B₃			
<hr/>					
S₅	S₄	S₃	S₂	S₁	S₀



برای دو عدد n بیتی به:

$$AND \Rightarrow 2 * (n * (n-1)) / 2 + n = n^2$$

$$Halfadder \Rightarrow n$$

$$Fulladder \Rightarrow (n-2)(n-1)/2 + 2(n-2) + (n-3)(n-2)/2 = (n^2 - 3n + 2 + 4n - 8 + n^2 - 5n + 6)/2 = n^2 - 2n$$

عملی:

متاسفانه برنامه بر روی سیستم بنده به درستی کار نمی‌کرد (ران و سیو مناسبی نمی‌کرد و در روند نصب و اجرای لایبری مشکل دارد) و درست کردن آن وقت بسیاری گرفت و در نهایت کاملاً درست نشد و نتوانستم در زمان معین به درستی همه را انجام دهم و بخش اول عملی کامل شد و بخش دوم متاسفانه کامل نشد (با استفاده از $dff\ 4$ و $mux\ 4$ خروجی هر ماکس را به ورودی dff وصل کرده و با توجه به جدول به mux ها ورودی می‌دهیم اگر موازی بود همان مقدار اگر شیفت بود بنا بر راست یا چپ بودن با قبلی یا بعدی و با استفاده از $serial_in$ ورودی می‌دهیم و خروجی ها را از dff ها می‌گیریم).