

درس معماری کامپیوتر

۱۴۰۴/۰۱/۱۸

تمرین سری سوم

۴۰۲۱۰۵۷۲۷

متن باقری

(۱) آ در اسلایدها داشتیم:

- K-Stage n-bit CSA
- Delay
– $(n/k) \cdot D_{FA} + (k-1) \cdot D_{mux}$

$$n = 32, \text{delay} = 32 \times 2 / k + 3 \times (k - 1) = 64/k + 3k - 3$$

مقدار این تابع به ازای عدد طبیعی ۵ کمینه است و از آنجا که میتوانیم اندازه بلوک ها را متفاوت در نظر بگیریم، ۳ بلوک ۶ بیتی و ۲ بلوک ۷ بیتی داریم:

$$\text{delay} = \max(7, 6) \times 2 + 4 \times 3 = 26$$

در این حالت تاخیر کمینه و برابر با ۲۶ نانوثانیه است.

ب) برای کمترین تاخیر، باید یکی از اعداد صفر باشد و تاخیر برابر می شود با $32 \times (3+2) = 160 \text{ ns}$

برای بیشینه شدن تاخیر، ضرب شونده باید یکی در میان ۰ و ۱ باشد و بیت سمت راست نیز ۱ باشد تا بیشترین تعداد تغییر

بین ۰ و ۱ های متوالی را داشته باشیم. تاخیر این حالت: $32 \times 27 +$ حالت قبل

$$32 \times 27 = 864 \text{ ns}$$

(۲) در جمع‌کننده‌ی Skip Carry، برای بهبود سرعت Skip کردن بیت carry انجام می‌شود. این روش Carry را در برخی موقعیت‌ها مستقیماً از ورودی به خروجی منتقل می‌کند. منطق این کار این است که در صورت ۱ شدن همه p_i ها carry (propagate carry) ورودی را بدون اینکه در محاسبات جمع دخیل کنیم، مستقیماً به carry خروجی بدهیم.

(ب) خیر، همانطور که توضیح داده شد بستگی به شرایط اعداد ورودی دارد و اگر شرط مورد نظر برقرار نباشد، مزیتی ندارد.

(ج) با تعریف P_i از اسلایدها آشنا هستیم، برای ۳ بیت داریم:

$$P_0 = A_0 \text{ xor } B_0, \quad P_1 = A_1 \text{ xor } B_1, \quad P_2 = A_2 \text{ xor } B_2,$$

تعریف می‌کنیم:

$$P = P_0.P_1.P_2$$

و خواهیم داشت:

$$C_{out} = C_{in} . P + \sim P . C_{internal}$$

$C_{internal}$ کرای است که با جمع کردن دو عدد A و B ساخته می‌شود و بدون در نظر گرفتن C_{in} است.

(د) به ازای ۲ بیت از ۲ عدد، در نصف حالات حاصل xor آنها ۱ است. پس در کل ۲ عدد n بیتی داریم و یعنی احتمال خواسته سوال 2^{-n} است.

(۳) (آ) چون اعداد ۶ بیتی اند، ۶ بار شیفت داده و ازین بابت $6*8=48$ تاخیر داریم.

$$\text{Sub delay} = 4 + 12 = 16 \text{ ns}$$

A	B	action	delay	To sum
101000	011001	sub	16	011000
101000	01100	Add	12	1010000

101000	0110	—		0
101000	011	Sub	16	011000000
101000	01	—		0
101000	0	Add	12	10100000000

Answer = 0011 1110 1000

A	B	action	delay	To sum
011001	101000	—	16	0
011001	10100	—	12	0
011001	1010	—		0
011001	101	Sub	16	100111000
011001	10	add		0110010000
011001	1	sub	12	10011100000
011001	—	add		

Answer = 0111 1010 1000

ب) تاخیر در دو حالت برابر است با $ns104$ چون هر دو عدد دو رشته از ۱ ها دارند و فرقی ندارد کدام را شیفیت دهیم. همچنین در این محاسبه فرض شده که توانایی ذخیره مکمل را نداریم و در هر مرحله باید تفریق باید آن را محاسبه کنیم.

(۴)

فرض کنید $n-1$ بیت را جمع کردیم، برای افزودن بیت دیگر به دو عدد تاخیر جدید می‌شود:

$$F(n) = f(n-1) + f(n-1)$$

همچنین برای جمع دو عدد ۱ بیتی تنها به یک جمع کننده کامل نیاز است و در ۱ زمان انجام می شود پس: $f(1) = 1$

$$f(n) = 2^{n-1}$$

(۵) آ می خواهیم چهار عدد باینری ۴ بیتی را با Carry Save Adder جمع کنیم (تعداد اعداد و بیت های آنها در این مثال به این صورت است و برای حالت های دیگر نیز روند کلی همین است). چهار عدد به این صورت اند:

$$A = a_3 a_2 a_1 a_0$$

$$B = b_3 b_2 b_1 b_0$$

$$E = e_3 e_2 e_1 e_0$$

$$F = f_3 f_2 f_1 f_0$$

در سطر بالا هر بلوک یک FA است که ۳ بیت را می گیرد و دو خروجی sum و carry تولید می کند.

در سطر بعدی، عدد A که از a_0 تا a_3 هست را به همراه خروجی های s' و c' از مرحله قبل وارد جمع کننده ها می کنیم. مجدداً در هر مرحله دو خروجی sum و carry جدید داریم.

در سطر سوم، جمع کننده ها بیت های حاصل از مرحله قبل را با هم جمع می کنند تا حاصل نهایی را به صورت s_0 تا s_5 تولید کنند.

این مدار به جای اینکه بیت به بیت و به صورت زنجیره ای carry را منتقل کند، به صورت موازی جمع ها را انجام داده و از چند لایه جمع کننده استفاده می کند.

این روش باعث افزایش سرعت محاسبه به ویژه در جمع چند عدد بزرگ می شود.

ب) (از آنجا که در سوال به محدودیت تعداد ورودی گیت‌ها اشاره‌ای نشده و همچنین می‌دانیم تاخیر گیت‌ها ۲ ورودی و ۳ ورودی اختلاف چندانی ندارد، فرض کردیم استفاده از گیت‌های ۳ ورودی نیز مجاز است. مثل گیت xor ۳ ورودی برای محاسبه s در FullAdder)

در ردیف آخر تصویر ۴ عدد FA داریم که به FA carry قبل از خود نیاز دارند، پس تاخیر این ردیف برابر با ۴ برابر تاخیر یک FA است.

اما با استفاده از CLA برای محاسبه carry مثلاً آخرین FA از این معادله استفاده می‌شود:

$$C3 = G2 + P2.G1 + P2.P1.G0 + P2.P1.P0.C0$$

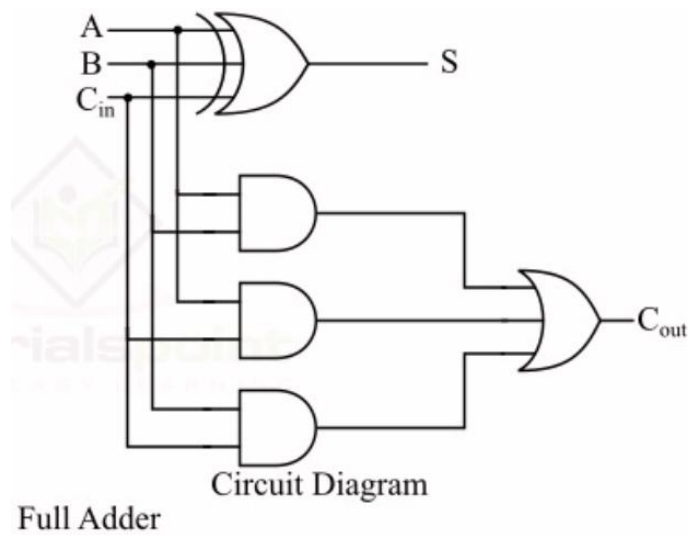
بنابراین تاخیر این حالت برابر است با تاخیر یک or به علاوه تاخیر ورودی‌های آن (مثلاً طولانی‌ترین جمله که هست

$$P2.P1.P0.C0 \text{ و معادل است با } (a0.b0) . (a0 \text{ xor } b0) . (a1 \text{ xor } b1) . (a2 \text{ xor } b2)$$

که در آن‌ها xor زمان‌بر تر از and است و سپس با هم and می‌شوند، پس تاخیر محاسبه carry‌های ردیف آخر برابر است با:

$$d_{OR} + d_{AND} + d_{XOR} = 2d$$

سپس این carry‌ها (به جز آخری) وارد FA می‌شوند. هر FA را به شکل زیر در نظر گرفتیم:



که تاخیر آن برابر است با d . بنابراین تاخیر ردیف پایین در مجموع برابر است با: $2d + d = 3d$

همچنین ۲ لایه FA در ۲ ردیف بالا داریم و تاخیر کل مدار می‌شود: $3d + 2d = 5d$