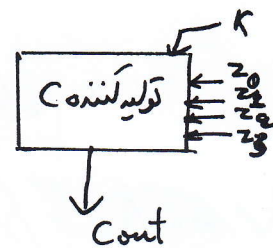
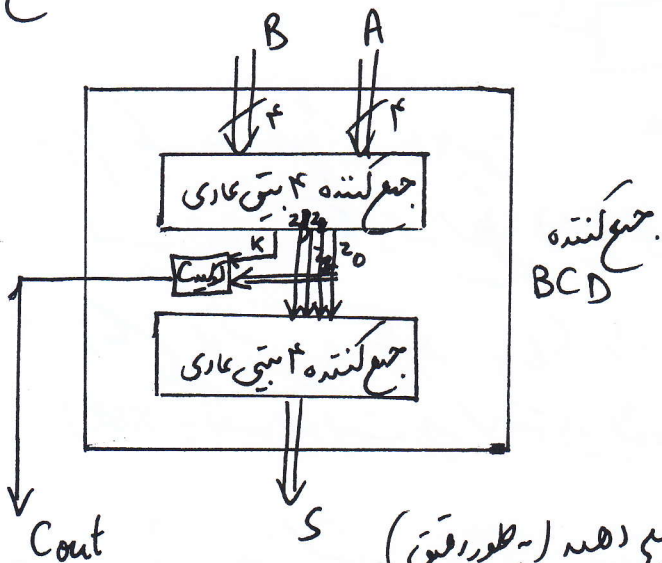


۱- ضرب را به روشی که گذاری بورت نمایش و انجام دهی.

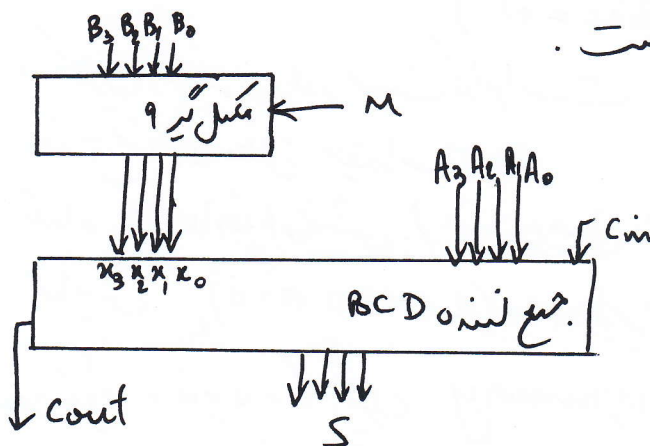
$$\begin{array}{r} + 25 = M \\ X - 13 = m \end{array}$$

۲-۱ تنها باید ROM که ابداً آن را تعیین می کنید و یک XNOR همین ضرب را چگونه باید انجام داد؟

۲-۱-۲ اگر A، B و S اعداد ۴ بیتی با نمایش BCD باشند (یعنی فقط مقادیر عددی صفر تا نه را بگیرند)، مدار منطقی تولید کننده C را به گونه ای به دست آورید که $(C, S) = A + B$ باشد (نشانگر انجام جمع دهد) (BCD) باشد.



۲-۲ با فرض داشتن یک جمع کننده BCD فوق، توضیح دهید (به طور دقیق) (یا مداره معادل) مدار مکمل گیر ۹ (9's complementer) چگونه کاری کند و مدار زیر یک جمع کننده/تولید کننده BCD است.



$$\begin{array}{ll} M = 0 & (C, S) = A + B \\ M = 1 & (C, S) = A - B \end{array}$$

(یعنی x_0, x_1, x_2, x_3 را چگونه ضرب B_3, B_2, B_1, B_0 و M باید به دست آورد که مکمل ۹ (۱۰-۱) B را داشته باشیم)

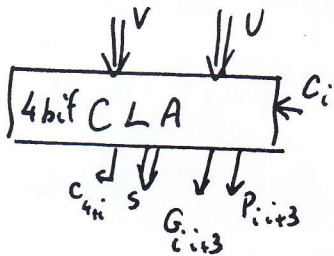
۳- اندازه M: حافظه اصلی 1 GByte، اندازه حافظه ثانویه: 1 MByte، اندازه هارد دیسک برابر 64 GByte و گذشت حافظه مستقیم است. میارین مختلف آدرس را تعیین کنید.

۳-۲. پس بجای آن با فرض اینکه گذشت حافظه شرکت پذیر مجبور است (set associative)

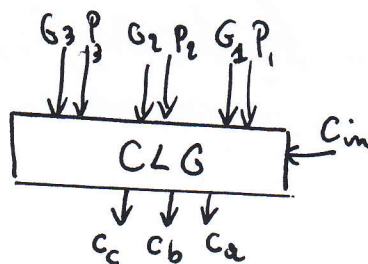
۲-۲ و هر مجموعه دارای هفت بلوک است پاسخ دهید.

۳-۳. نرخ برخورد (hit ratio) حافظه برای برابر ۹۴٪، $t_m = 50ns$ و $t_c = 10ns$

۱-۱ زمان متوسط دسترسی t_{avg} به سیستم حافظه را می‌توانید بگویند.



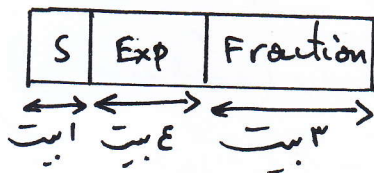
۴-۱. فرض کنید این بلوک ۴ می‌تواند (جمع) را در اختیار دارید:



CLA: Carry Lookahead Adder
CLG: Carry Lookahead Generator

۴-۱. طرح یک مدار جمع کننده ۴۴ بیتی
۲-۲. آیا یک بلوک ۴ و سه سطح (سطح اول CLA ۴ و سطح دوم و سوم CLG ۴) بسازید.

۴-۲. زمانیکه می‌تواند S_{63} و C_{64} را با فرض تأخیر یک Δ برای یک پایه می‌توانید بسازید.
۲-۲ (XOR: یک چندبیتی پایه ساخته باید بود)



۵-۱. در نمایش منجر شده با قالب مقابل هفده:

$$N = (-1)^S \cdot 1.F \times 2^E$$

(Bias = +7)

۲-۱. ۵-۱. کوچکترین و بزرگترین عدد مثبت کدام است؟

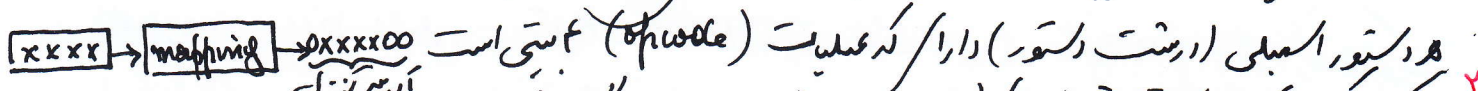
۲-۲. ۵-۲. دقت نسبی و مطلق چیست است؟

۱-۳. ۵-۳. مقدار کل اعداد قابل نمایش (مثبت، منفی، صفر) چند است؟

۱-۴. ۵-۴. نمایش $N = (10110010)_2$ برابر چه عددی است؟

۴-۲. ۶-۱. بایک مدل، فایده به کارگیری Nanomemory را در رمزبرنامه نویسی نشان دهید.

۷-۱. فرض کنید حافظه رمزبرنامه کنترل یک برآزنه دارای $n \times n$ بیتی است (n طول کلمه کنترل است)



۲-۲. که یک آدرس ۷ بیتی تبدیل می‌شود. شرح دهید برآزنه چگونه کلمات کنترلی را برای درخت درخت

و درخت درخت و والستی و بقیه موارد اجزای کنده و آرایشی احتمالی حافظه کنترلی چگونه است. مدون بایست
۲۰

تصمیم امکان پایداری نرم مکتبی کامپیوتر (۳۲۴-۴) ۱۳۹۲، ۳، ۲۲

به هم خدا

۲۷

$$M = +25 = 011001$$

۱-۱- با ۶ و ۱۳- با ۵ بیت قابل نمایش هستند:

$$m = -13 = C_2(01101) = 10011$$

$$-10 + 10 - 1 = -1$$

$$-2^4 + 2^2 - 2^0 = -16 + 4 - 1 = -13 \quad \checkmark$$

نکته: ۱۰ بیت برای نمایش حاصل ضرب کافی است که بیت
سیست بیتی و محصولات عددها خواسته بود

$$M = 011001$$

$$m = \bar{1}0101$$

Sign Ext

$$1111100111 = -M = (-25)$$

$$0000000000 = 0M$$

$$00011001 = 1M$$

$$00000000 = 0M$$

$$100111 = -M$$

$$1010111011 = P$$

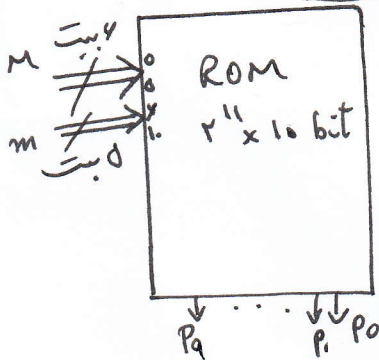
$$2^8 + 2^7 + 2^2 + 2^0$$

$$= 0101000101$$

$$= 256 + 128 + 4 + 1 = 389$$

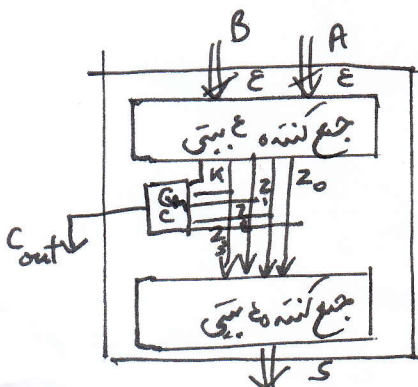
$$\Rightarrow P = -389 = -13 \times 25$$

۲-۱ از آنجا که ضرب یک عمل سریایی است می توان با یک حافظه (مثل ROM) آنرا می گوییم
نمودار. کافی است M و m را به عنوان خطوط آدرس به حافظه عرضه کنیم. M دارای ۶ و m دارای ۵ و حاصل ضرب ۱۰
بیت است.



راه حل اول: M و m را مستقیماً به خطوط آدرس وصل کنیم
و با آدرس دهی به خانه ۱ که از قبل مقداردهی شده حاصل ضرب
۱۱ بیتی را بخوانیم. چون خطوط آدرس ۱۱ بیت است پس
۲ خانه ۱۱ بیتی داریم. در مورد این ضرب خاص،
حاصل ضرب در آدرس
(۱۰۰، ۱۱۰، ۱۱۰۰، ۱۱۰۰۱)_۲
(4 D 9)_{۱۶} قرار دارد.

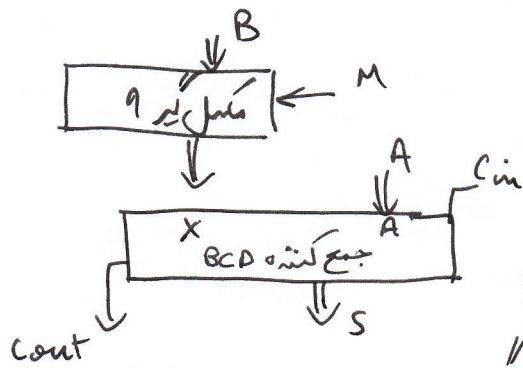
راه حل دوم: اگر علامت M و m (مثل حالت فوق) ثابت باشد یعنی M همیشه مثبت باشد نیازی به
۴ بیت آدرس برای M نیست و حافظه را می توان به ابعاد ۱۰ x ۱۰ بیت رفت و بیت علامت
را با XOR بیت علامت M و m به دست آورد.



$$C_{out} = K + Z_3(Z_2 + Z_1)$$

۲-۱ جمع کننده BCD

(ص ۱)



۲-۲ - $M=1, C_{in}=1$ تفريق
 $M=0$ جمع

$$A-B = A + C_{10}(B) = A + C_9(B) + 1$$

برای انجام تفريق، کافی است مکمل ۱۰ عدد B را بگیریم.
 این کار را می توانیم با گرفتن مکمل ۹ + اضافه کردن یک
 از طریق مینوس انجام دهیم. حال ببینیم مکمل ۹ اعداد
 BCD کدام است:

B	b_3	b_2	b_1	b_0	x_3	x_2	x_1	x_0	
0	0	0	0	0	9	1	0	0	1
1	0	0	0	1	8	1	0	0	0
2	0	0	1	0	7	0	1	1	1
3	0	0	1	1	6	0	1	1	0
4	0	1	0	0	5	0	1	0	1
5	0	1	0	1	4	0	1	0	0
6	0	1	1	0	3	0	0	1	1
7	0	1	1	1	2	0	0	1	0
8	1	0	0	0	1	0	0	0	1
9	1	0	0	1	0	0	0	0	0

— راه حل منطقی (ریاضی):
 x_3, x_2, x_1, x_0
 را حسب b_3, b_2, b_1, b_0 و b_3 و b_2 و b_1 و b_0 از طریق جدول کاروف
 بدست آوریم:

$$\begin{cases} x_0 = \bar{b}_0 \\ x_1 = b_1 \\ x_2 = b_1 \oplus b_2 \\ x_3 = \bar{b}_3 \bar{b}_2 \bar{b}_1 \end{cases}$$

— راه حل دوم: استفاده از یک ROM یا Encoder

— راه حل سومی: کافی است همه بیت های B را مکمل بگیریم و با مقدار عددی ۱۰ جمع کنیم و بیت نقلی
 (carry) را نادیده بگیریم: مثال:

$$B = 3 = 0011 \rightarrow \bar{B} = 1100 \quad \bar{B} + 10 = (22)_{10} = (10110)_2$$

$$C_9(B) = 6$$

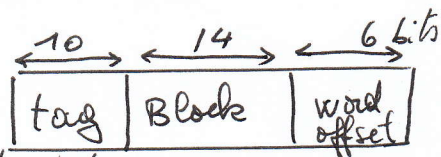
— راه حل سومی: عدد ۶ را به B اضافه کنیم و پس مکمل بیت به بیت بگیریم:

$$B = 3 \rightarrow B + 6 = 9 = 1001 \rightarrow 0110 = 6 = C_9(B)$$

$$\begin{aligned} MM &= 1GB = 2^{30}B = 2^{24} \text{ Blocks} \\ \text{Cache} &= 1MB = 2^{20}B = 2^{14} \text{ Blocks} \\ 1 \text{ Block} &= 64 \text{ Byte} = 2^6 B \end{aligned}$$

Direct mapping

CPU Address:

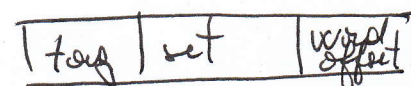


Block i in Main Memory \leftrightarrow Block $(i \bmod 2^{14})$ in cache

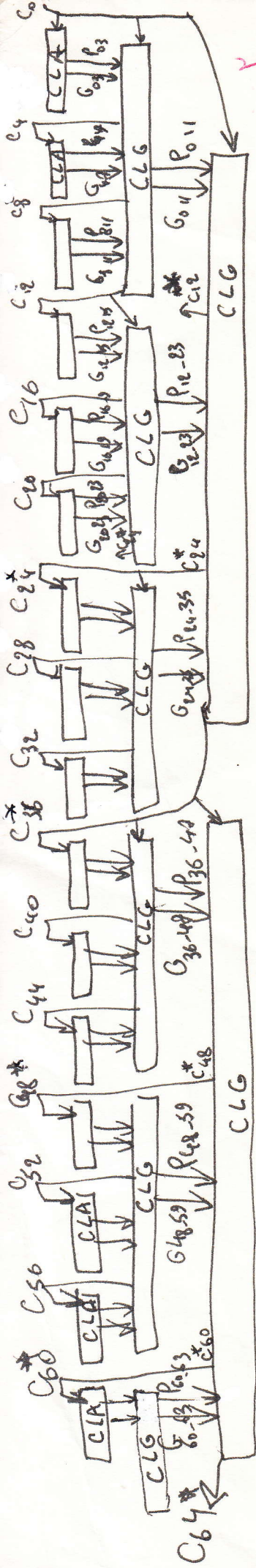
$$1 \text{ set} = 8 \text{ Blocks} = 2^3 \text{ Blocks}$$

$$\text{Cache} = 1MB = 2^{20}B = 2^{14} \text{ blocks} = 2^{11} \text{ set}$$

Block i in MM \leftrightarrow one block in set $(i \bmod 2^{11})$ in cache
 (selected associatively)



$$t_{av} = h t_c + (1-h) t_{MM} \Rightarrow (20) \quad t_{av} = 0.94 \times 10 + 0.06 \times 50 = 12.4 \text{ ns}$$



دالة الجزي

$$A_i, B_i, C_i \rightarrow 0$$

$$P_j, G_i \rightarrow \Delta$$

$$P_{j+3} \rightarrow 2\Delta$$

$$P_j, P_{j+3}, C_{in}, G_{i+3} \rightarrow 3\Delta$$

$$P_{i+11}, C_4 \rightarrow 4\Delta$$

$$G_{i+11}, C_8, C_{12} \rightarrow 5\Delta$$

$$C_{12}^* \rightarrow 6\Delta$$

$$C_{24}^*, C_{36}^* \rightarrow 7\Delta$$

$$C_{48}^*, C_{60}^*, C_{64}^* \rightarrow 9\Delta$$

$$C_{61}, C_{62}, C_{63} \rightarrow 11\Delta$$

$$\Rightarrow S_{63} = 14\Delta$$

$$C_4 = G_{03} + P_{03} C_0$$

$$C_8 = G_{47} + P_{47} G_{03} + P_{47} P_{03} C_0 = G_{07} + P_{07} C_0$$

$$C_{12} = G_{811} + P_{811} G_{47} + P_{811} P_{47} G_{03} + P_{811} P_{47} P_{03} C_0$$

$$C_{12}^* = G_{011} + P_{011} C_0$$

$$C_{24}^* = G_{1223} + P_{1223} G_{011} + P_{1223} P_{011} C_0$$

$$C_{36}^* = G_{2435} + P_{2435} G_{1223} + P_{2435} P_{1223} G_{011} + P_{2435} P_{1223} P_{011} C_0$$

$$C_{16} = G_{1215} + P_{1215} C_{12}$$

$$C_{48}^* = G_{3648} + P_{3648} C_{36}^*$$

$$C_{60}^* = G_{4859} + P_{4859} G_{3648} + P_{4859} P_{3648} C_{36}^*$$

$$C_{64}^* = G_{6063} + P_{6063} G_{4859} + P_{6063} P_{4859} G_{3648} + P_{6063} P_{4859} P_{3648} C_{36}^*$$

