سیاوش کاوسی - ۹۲۳۱۰۴۸

تمرین اول ریزپردازنده

۱-

نام پردازنده

TMS320C82 DIGITAL SIGNAL PROCESSOR

- انجام بیش از ۱.۵ میلیارد دستور ریسک
- پردازنده اصلی
  - پردازنده با معماری ریسک ۳۲ بیتی
  - محاسبات شناور IEEE-754
  - حافظه نهان دستورات 4K-Byte
  - حافظه نهان داده 4K-Byte

- دو پردازنده موازی
  - پردازنده ۳۲ بیتی پردازش تصویر پیشرفته
  - کد دستور ۶۴ بیتی با پشتیبانی از انجام چند دستور همزمان
  - حافظه نهان دستورات 4K-Byte
  - مموری پارامتری 4K-Byte
  - مموری داده 8K-Byte

- کنترلر انتقال
  - انتقال داده ۶۴ بیتی
  - نرخ انتقال داده تا ۴۸۰ مگابایت بر ثانیه
  - آدرس دهی ۳۲ بیتی
  - Direct EDO DRAM/VRAM Interface
  - Direct SDRAM Interface
  - سایز باس متغیر
  - صف بندی و اولویت سیکل هوشمند

- عملیات های big-endian یا little-endian
- ۴۴K رم (RAM) بر روی تراشه
- فضای آدرس دهی ۴ گیگابایت
- زمان یک سیکل ۱۶.۶ نانوثانیه
- (IEEE 1149.1 Test Port (JTAG

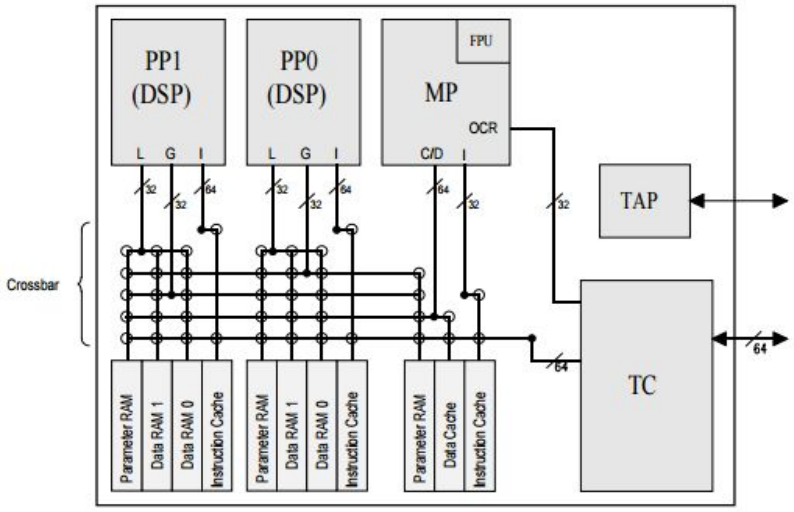

Figure 1. 'C82 Block Diagram Showing Datapaths

بلوک مربوط به دو پردازنده پردازش سیگنال

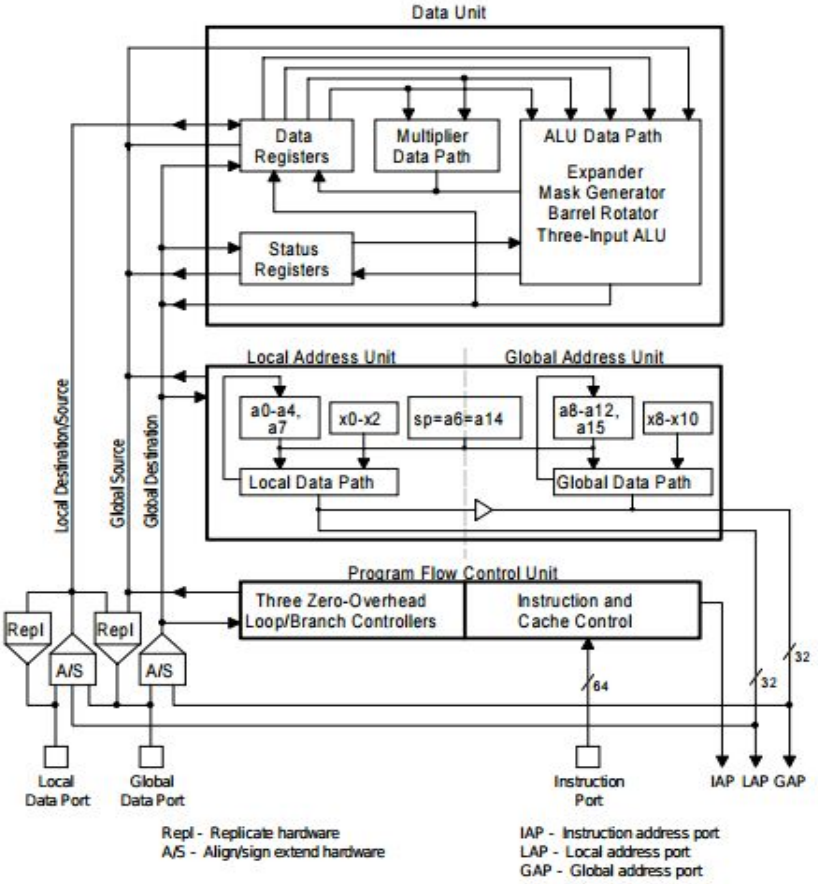

Figure 22. PP Block Diagram

ب)

دستور **divi**
انجام یک بار تکرار الگوریتم تقسیم بدون علامت. بدست آوردن یک بیت خارج قسمت به ازای هر اجرا با استفاده از تکرار تفریق

قواعد

dst1 = [cond [.pro]] divi (src2, dst2 = cond src1 [n src1-1])

مثال

d3 = divi (d1, d2 = d2)
d3 = divi (d1, d2 = d3[n]d2)

دستور **MPY || ADD**
انجام ضرب ماتریسی ۱۶*۱۶ با عملیات جمع یا تفریق به صورت اختیاری. شرط ها به هر دو عملیات ضرب و جمع (تفریق)

قواعد

dst2 = [sign] [cond] src3 * src4 [|| dst = [cond [.pro]] src2 + src1 [n src1 -1]]
dst2 = [sign] [cond] src3 * src4 [|| dst = [cond [.pro]] src2 - src1 [n src1 -1]]

مثال

d7 = u d6 * d5 || d5 = d4 - d1

دستور **MPY || EALU**
Perform a multiply and an optional parallel EALU. Multiply can use rounding, scaling, or splitting features
انجام یک ضرب و یک عملیات اختیاری EALU همزمان. ضرب میتواند از گرد کردن, بزرگ و کوچک کردن, یا بخش کردن استفاده کند

قواعد

- فرم کلی

dst2 = [sign] [cond] src3 * src4 || dst = [cond [.pro]] ealu[f] (src2, src1 [n src1 -1] \\ d0, %d0)
dst2 = [sign] [cond] src3 * src4 || ealu()

- فرم صریح

dst2 = [sign] [opt] [cond] src3 * src4 [<<dms] || dst1 = [fmod] [cond [.pro]] ealu (label: (EALU_EXPRESSION
dst2 = [sign] [opt] [cond] src3 * src4 [<<dms] || ealu (label

مثال

- فرم کلی

d7 = [p] d5 * d3 || d2 = [p] ealu(d1, d6\\d0, %d0)

- فرم صریح

d2 = m d4 * d7 || d3 = ealu (mylabel: d3 + d2 >> 9)

۲-

★ سیستم های Lockstep سیستم هایی هستند که fault-tolerant هستند که برای اینکار مجموعه دستورالعمل های یکسانی به صورت همزمان اجرا می شوند. افزونگی(منظور افزایش تعداد پردازنده ها یا هسته ها) باعث تشخیص و تصحیح خطا می شود بدین صورت که اگرحداقل دو سیستم داشته باشیم (dual modular redundancy) خروجی عملیات های lockstep با یکدیگر مقایسه می شود و اگر خطایی بود به صورت خودکار تصحیح می شود اما در صورتی که سه سیستم داشته باشیم (triple modular redundancy) با رای گیری.

★ ECC-Protected Cache به حافظه نهانی که می تواند بیشتر خرابی های رایج داده را تشخیص و تصحیح کند. تکنیک EDC/ECC در حافظه های نهان خطا را در سطح اول حافظه تشخیص می دهد و در سطح دوم آن را تصحیح میکند. تکنیک ECC/ECC در هردو سطح از سیستم تصحیح خطا استفاده میکند.

۳- گزینه های ب

۴-

Physical Address = AAC0EH
SS = A980H
X (or BP) = AAC0EH - A9800H = 140EH
Logical Address = A980:140E

۵-

SI = 5000H
DS = 4B00H (I think it’s a mistake, Data segment should be 4-digits)

Logical Address = 4B00:5000
Physical Address = 4B000H + 5000H = 50000H
The lower range = 4B000H
The upper range = 5AFFFH