

## به نام خدا

ساختار کامپیوتر (۲۵۷۵۴)

امتحان پایان ترم

۱۳۹۴/۱۰/۲۶

۱. ماشین حالت (State Machine) پیاده‌سازی Multi-Cycle پردازنده MIPS که در متن درس مورد بحث قرار گرفت، عیناً در برگه ضمیمه آورده شده است. اتصالات و مالتی‌پلکسرهای مورد نیاز ساختار مسیره‌داده (Data Path) را در پیش‌نویس ارائه شده رسم و آن را تکمیل نمایید. (۱۵ نمره)

۲. فرض کنید دستور جدیدی با نام lwinc و با عملکرد زیر به مجموعه دستورات پردازنده اضافه شود:

$$\text{lwinc } rt, rs, \text{imm} \quad \# \text{ } rt \leftarrow \text{Mem}[rs + \text{sign\_extend}(\text{imm})], \text{ } rs \leftarrow rs + 4$$

تغییر مورد نیاز در مسیره‌داده و کنترلر پیاده‌سازی Multi-Cycle برای اجرای این دستورالعمل را مشخص نمایید. (۱۰ نمره)

۳. برای اجرای دستور جدید lwinc که در بالا معرفی شد، چه تغییراتی بایستی در منابع (Resource) پیاده‌سازی Single-Cycle انجام گیرد؟ (۷ نمره)

۴. یک مدار جمع‌کننده در نظر بگیرید که دو عدد ۶۴ بیت را در ورودی خود دریافت نموده و ۶۵ بیت (شامل ۶۴ بیت حاصل جمع و یک بیت نقلی) را در خروجی محاسبه می‌نماید. با در اختیار داشتن جمع‌کننده‌های کامل ۱۶ بیتی با تاخیر  $1\text{ ns}$  ( $10^{-9}\text{ sec.}$ ) و رجیسترهایی که تاخیرهای آنها صفر است، مدار پایپ‌لاین چهار مرحله‌ای طراحی کنید که Throughput آن  $1\text{ G}$  ( $10^9$ ) Operation/Second باشد. (۱۵ نمره)

۵. یک Cache با مشخصات ذیل را در نظر بگیرید:

CPU Address Space:	$2^{24}$ words
Cache Size:	$4096$ ( $2^{12}$ ) words
Block (Line) Size:	8 words
Mapping:	8-way set associative
Write Strategy:	Write-Back

الف) بلوک دیاگرام این حافظه را رسم نمایید. (۸ نمره)

ب) تعداد بیت‌های Tag و محتوای هر فیلد آن را دقیقاً تعیین کنید. (۷ نمره)

ج) جهت پیاده‌سازی این Cache چه مقدار حافظه سریع مورد نیاز می‌باشد؟ (۳ نمره)

د) اگر آدرسهای زیر از چپ به راست توسط پردازنده تولید شوند، کدام Hit و چه تعدادی Miss خواهند شد؟ (۳ نمره)

4, 5, 6, 7, 8, 9, 10, 11, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

۶. قطعه کد مقابل را در نظر بگیرید. برای اجرای صحیح این قطعه کد در پایپ‌لاین پردازنده، کنترلر بایستی در دو حالت زیر بعد از کدام دستورات و به چه تعداد bubble قرار دهد؟ (۱۰ نمره)

```
add $t0, $s0, $s1
sub $t0, $t0, $s2
lw $t1, 60($t0)
and $t2, $t1, $t0
```

الف) پایپ‌لاین جز در Register File دارای هیچ قابلیت data forwarding نیست.

ب) کلیه data forwarding های ممکن در ساختار پایپ‌لاین پیاده‌سازی شده اند.

۷. فرض کنید مقایسه مقادیر رجیسترها برای اجرای دستورات پرش (beq و bne) به جای استفاده از ALU موجود در مرحله EXE، در مرحله DE (دیکود) توسط یک مقایسه‌کننده مستقل انجام شود.

الف) در صورت وابستگی رجیسترهایی که در یک دستور پرش خوانده می‌شوند به رجیسترهایی که در دستورات قبل آن نوشته شده‌اند، بایستی چه اقدامی صورت پذیرد؟ این اقدامات را در کلیه حالت‌های ممکن بصورت دقیق مشخص کنید. (۱۵ نمره)

ب) با فرض فوق آیا ممکن است delay slot دستورات پرش فقط به یک دستور و یا دو دستور محدود گردد؟ چگونه؟ (۷ نمره)

## State Machine

- Reset:
  - $MRE \leftarrow 1$
  - $MAR \leftarrow PC$
  - Goto Fetch1
- Fetch1:
  - Goto Fetch2
- Fetch2:
  - Goto Fetch3
- Fetch3:
  - $MRE \leftarrow 0$
  - $PC \leftarrow PC + 4$
  - $IR \leftarrow \text{Mem. Read Data}$
  - Goto Decode

Reminder: Memory Address to Read-Data propagation delay is 7 ns, i.e. 3 cycles  
 $PC \leftarrow PC + 4$  can occur in any of Fetch1, Fetch2 or Fetch3

## State Machine, cont.

- Decode:
  - $A \leftarrow \text{RegFile}[IR[25:21]]$
  - $B \leftarrow \text{RegFile}[IR[20:16]]$
  - On  $IR[31:26, 5:0]$ , Goto:
    - Exec-alu-R
    - Exec-alu-l
    - Exec-lw-1
    - Exec-sw-1
    - Exec-beq-1
- Exec-alu-R:
  - $\text{RegFile}[IR[15:11]] \leftarrow A \otimes B$
  - \$Prepare-Fetch
- Exec-alu-l:
  - $\text{RegFile}[IR[20:16]] \leftarrow A \otimes \text{Sgn/Zro-Ex}(IR[15:0])$
  - \$Prepare-Fetch

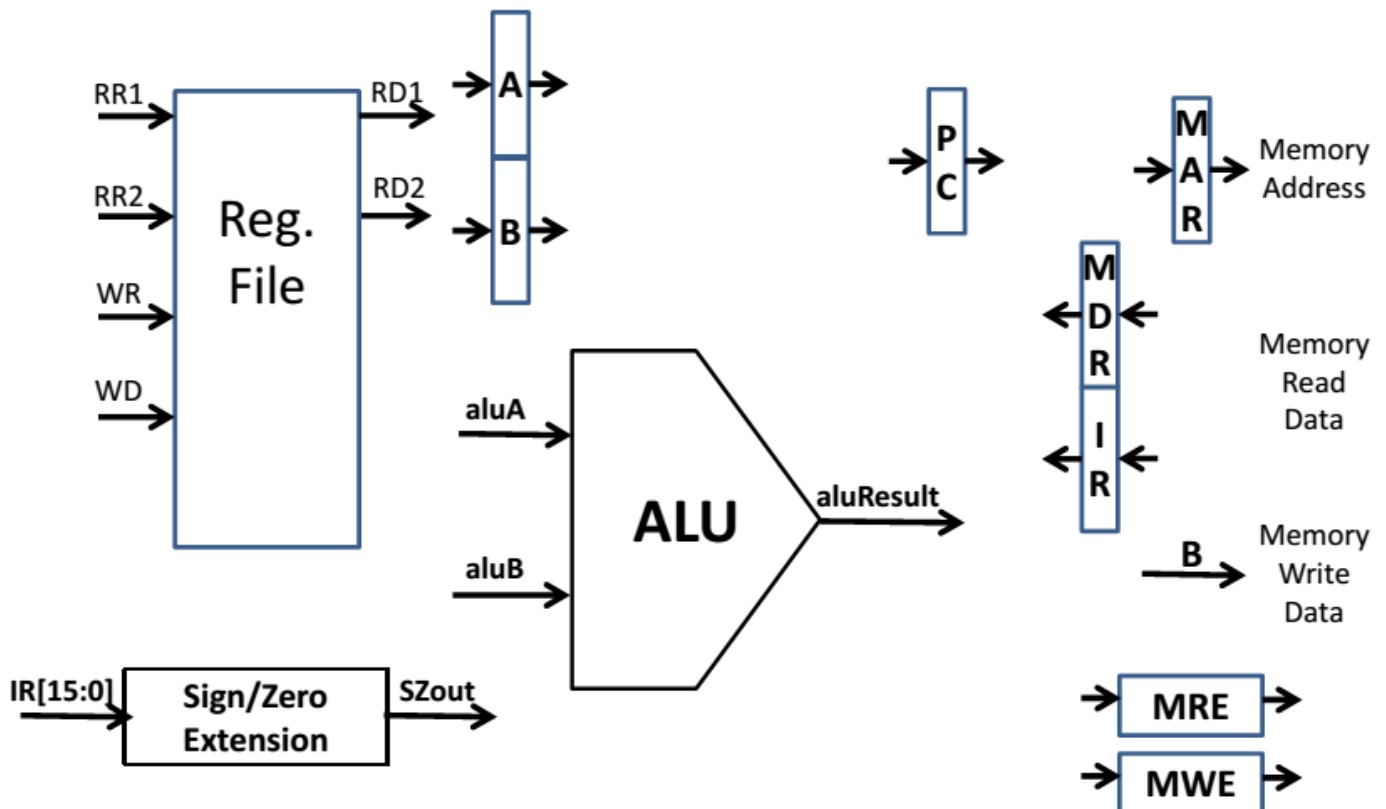
## State Machine, cont.

- Exec-lw-1:
  - $MAR \leftarrow A + \text{SgnEx}(IR[15:0])$
  - $MRE \leftarrow 1$
  - Goto Exec-lw-2
- Exec-lw-2:
  - Goto Exec-lw-3
- Exec-lw-3:
  - Goto Exec-lw-4
- Exec-lw-4:
  - $MRE \leftarrow 0$
  - $MDR \leftarrow \text{Mem. Read Data}$
  - Goto Exec-lw-5
- Exec-lw-5:
  - $\text{RegFile}[IR[20:16]] \leftarrow MDR$
  - \$Prepare-Fetch

## State Machine, cont.

- Exec-sw-1:
  - $MAR \leftarrow A + \text{SgnEx}(IR[15:0])$
  - $MWE \leftarrow 1$
  - Goto Exec-sw-2
- Exec-sw-2:
  - $MWE \leftarrow 0$
  - Goto Exec-sw-3
- Exec-sw-3:
  - \$Prepare-Fetch
- Exec-beq-1:
  - $A - B$
  - if (aluZero)
    - Goto Exec-beq-2
  - else
    - \$Prepare-Fetch
- Exec-beq-2:
  - $PC, MAR \leftarrow PC + \text{SgnEx}(IR[15:0]) \times 4$
  - $MRE \leftarrow 1$
  - Goto Fetch1

# Data Path, Final Draft



MAR: Memory Address Register, MDR: Mem. Data Reg., MRE: Mem Read Enable, MWE: Mem. Write Enable