

פסיקות *Exception* *Interrupt*

ארגון המחשב ושפת סף

מרצה: רועי אש

אפקה

המכללה האקדמית
להנדסה בתל אביב



Exceptions/Interrupts

(MIPS terminology)

- Exception – an unexpected event from within the processor
- Interrupts – an event that causes an unexpected change in control flow but comes from outside of the processor.

חריגות (exception) ופסיקות (interrupts)

- אחד הנושאים הקשים ביותר ליישום בבקרה הוא חריגות (exceptions) ופסיקות (interrupts) – אירועים המשנים את הזרימה הסדרתית של ביצוע הפקודות בתכנית, ואינם branch או jump. האירועים האלה נוצרו כדי לטפל במקרים לא צפויים בתוך המעבד, כמו גלישה, והורחבו כדי לכסות את הטיפול בתקשורת בין המעבד ובין יחידות קלט/פלט. יש ארכיטקטורות שמבחינות בין סוגי אירועים על-פי גורמיהם, פנימיים או חיצוניים. המוסכמה בשפה של ה-MIPS היא שהמושג חריגה כולל את כל סוגי האירועים, והמושג פסיקה מתייחס רק לאירוע שהגורם לו הוא חיצוני.
- מנגנון הפסיקה הוא מנגנון המשלב חומרה ותוכנה הנועד לטפל באירועים המוגדרים כפסיקות או חריגות

Interrupt and exception

Type of event	From where?	MIPS terminology
I/O device request	External	Interrupt
Invoke the operating system from user program	Internal	Exception
Arithmetic overflow	Internal	Exception
Using an undefined instruction	Internal	Exception
Hardware malfunctions	Either	Exception or interrupt

MIPS exceptions facility

- MIPS exception facility responds both to exceptions caused by errors during an instruction's execution and to external interrupts caused by I/O devices.
- In MIPS processors, a part of the CPU called *coprocessor 0* records the information that software needs to handle exceptions and interrupts.
- The MIPS simulator MARS implement some of coprocessor 0's registers.



Mips coprocessor 0's registers

Register name	Usage
Cause	exception type and pending interrupt bits.
EPC (Exception PC)	Address of next instruction to that caused exception (PC+4)

קיימים עוד אוגרים ועוד פרטים המתייחסים
ל- coprocessor 0 אנחנו בקורס נתייחס
רק ל- Cause ו- EPC



Exceptions handling

Interrupt handler

(ISR - *Interrupt Service Routine*)

A piece of code that is run as a result of an exception or an interrupt.

We have 2 ways to handle exceptions:

- Cause register
- Vectored interrupts

MIPS – Cause register



MIPS OS Exception Handler

- Exceptions and interrupts cause a MIPS processor to jump to a piece of code, at address 0x80000180 (in the kernel, not user address space), called an exception handler.
- This code examines the exception's cause and jumps to an appropriate point in the operating system. The operating system responds to an exception either by terminating the process that caused the exception or by performing some action

Intel – 80 x 86 terminology

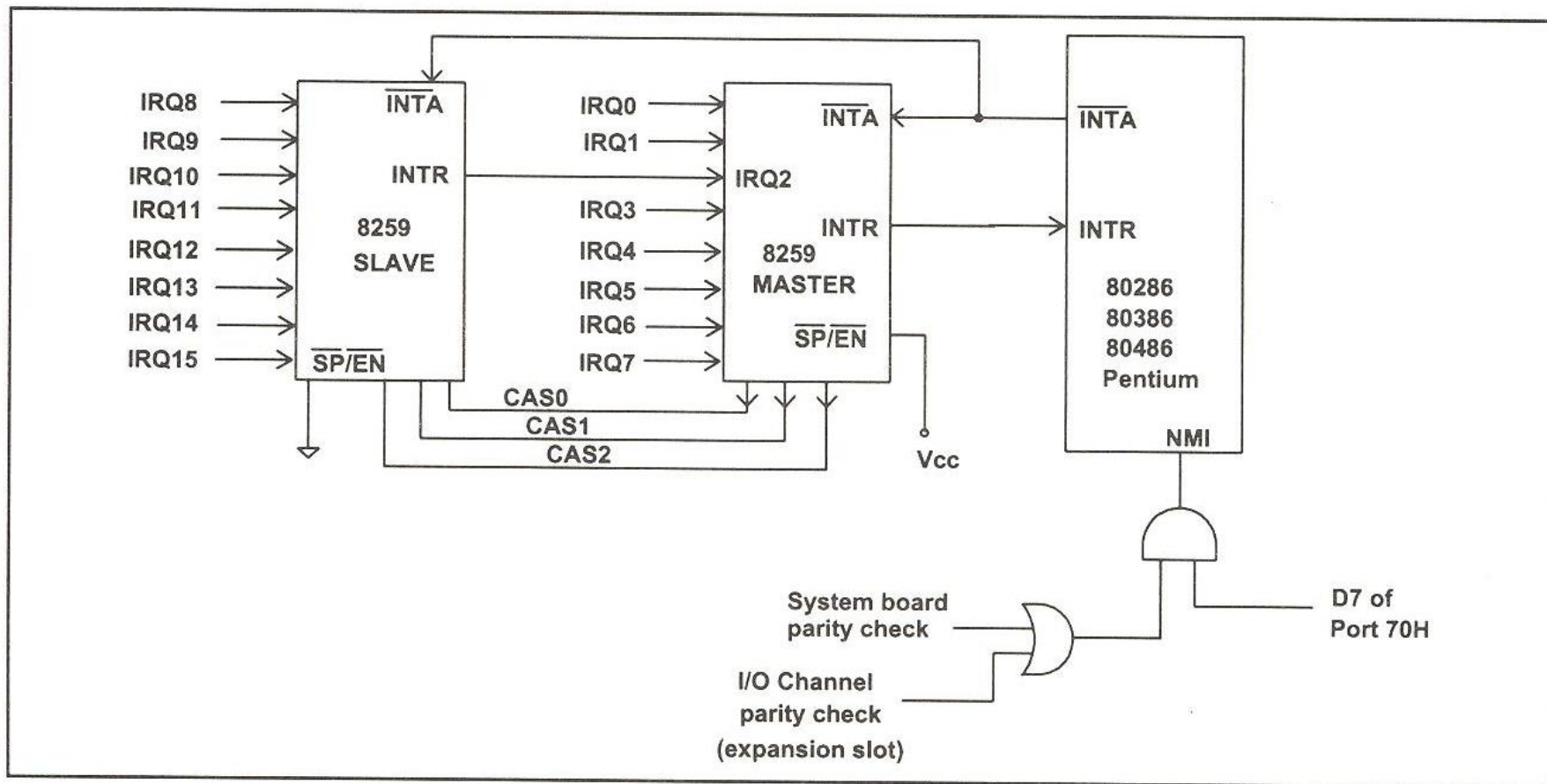
טבלת הפסיקות (IVT)

INT (Hex)	IRQ	Common Uses
00 - 01	Exception Handlers	-
02	Non-Maskable IRQ	Non-Maskable IRQ (Parity Errors)
03 - 07	Exception Handlers	-
08	Hardware IRQ0	System Timer
09	Hardware IRQ1	Keyboard
0A	Hardware IRQ2	Redirected
0B	Hardware IRQ3	Serial Comms. COM2/COM4
0C	Hardware IRQ4	Serial Comms. COM1/COM3
0D	Hardware IRQ5	Reserved/Sound Card
0E	Hardware IRQ6	Floppy Disk Controller
0F	Hardware IRQ7	Parallel Comms.
10 - 6F	Software Interrupts	-
70	Hardware IRQ8	Real Time Clock
71	Hardware IRQ9	Redirected IRQ2
72	Hardware IRQ10	Reserved
73	Hardware IRQ11	Reserved
74	Hardware IRQ12	PS/2 Mouse
75	Hardware IRQ13	Math's Co-Processor
76	Hardware IRQ14	Hard Disk Drive
77	Hardware IRQ15	Reserved
78 - FF	Software Interrupts	-

בקר פסיקות חומרה

PIC (Programmable Interrupt Controller) - 8259

מבנה 8259 עבור intel 80x286 ומעלה



איך זה עובד ב- *pipeline*?

אפקה

המכללה האקדמית
להנדסה בתל אביב

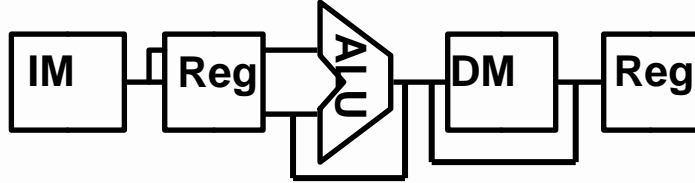


Dealing with Exceptions

- Exceptions (aka interrupts) are just another form of control hazard. Exceptions arise from
 - R-type arithmetic overflow
 - Trying to execute an undefined instruction
 - An I/O device request
 - An OS service request (e.g., a page fault, TLB exception)
 - A hardware malfunction
- The pipeline has to stop executing the offending instruction in midstream, let all prior instructions complete, flush all following instructions, set a register to show the cause of the exception, save the address of the offending instruction, and then jump to a prearranged address (the address of the exception handler code)
- The software (OS) looks at the cause of the exception and “deals” with it



Where in the Pipeline Exceptions Occur



- Arithmetic overflow
- Undefined instruction
- TLB or page fault
- I/O service request
- Hardware malfunction

Stage(s)?

Synchronous?

EX

yes

ID

yes

IF, MEM

yes

any

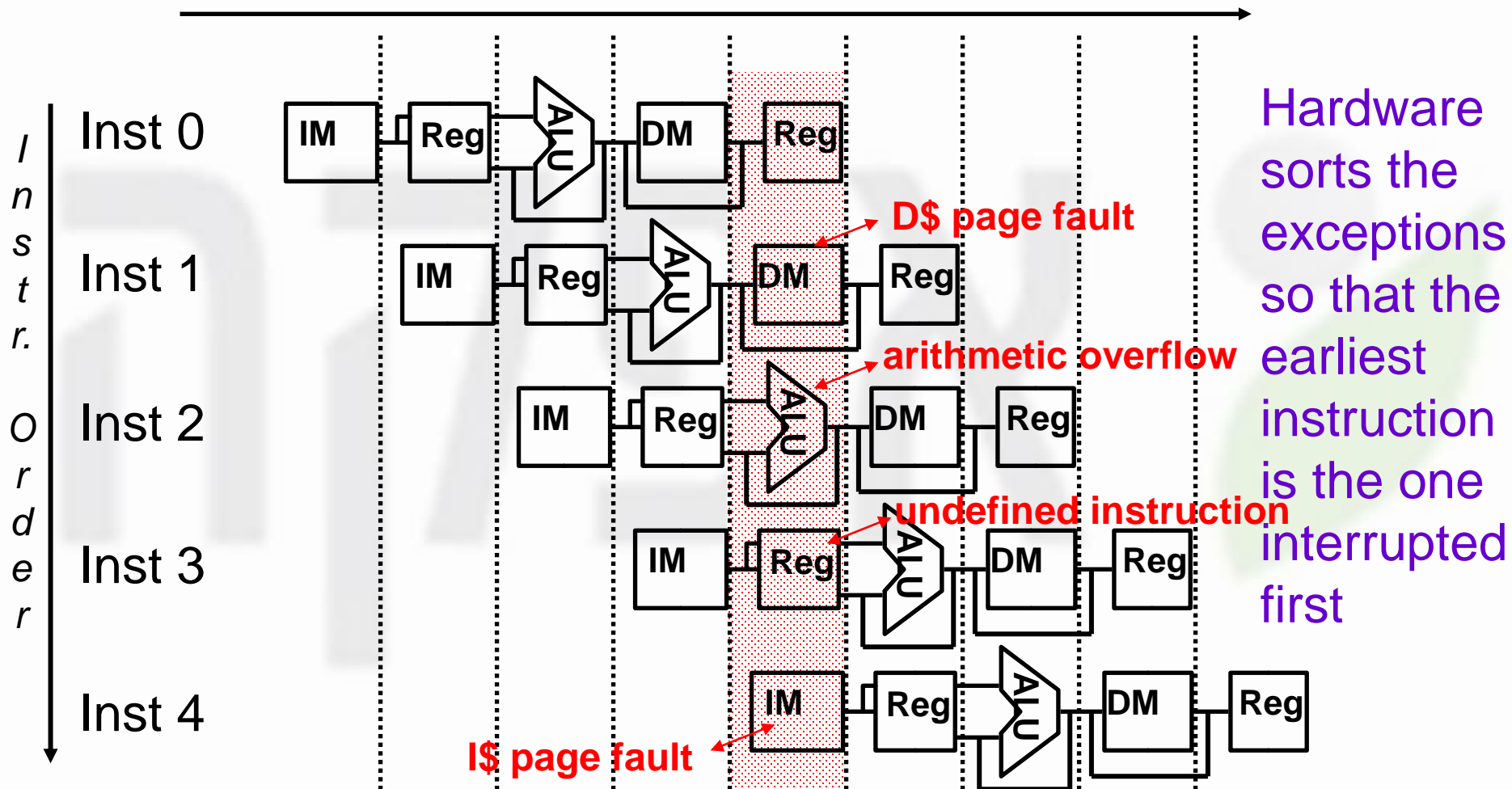
no

any

no

- ❑ Beware that multiple exceptions can occur simultaneously in a *single* clock cycle

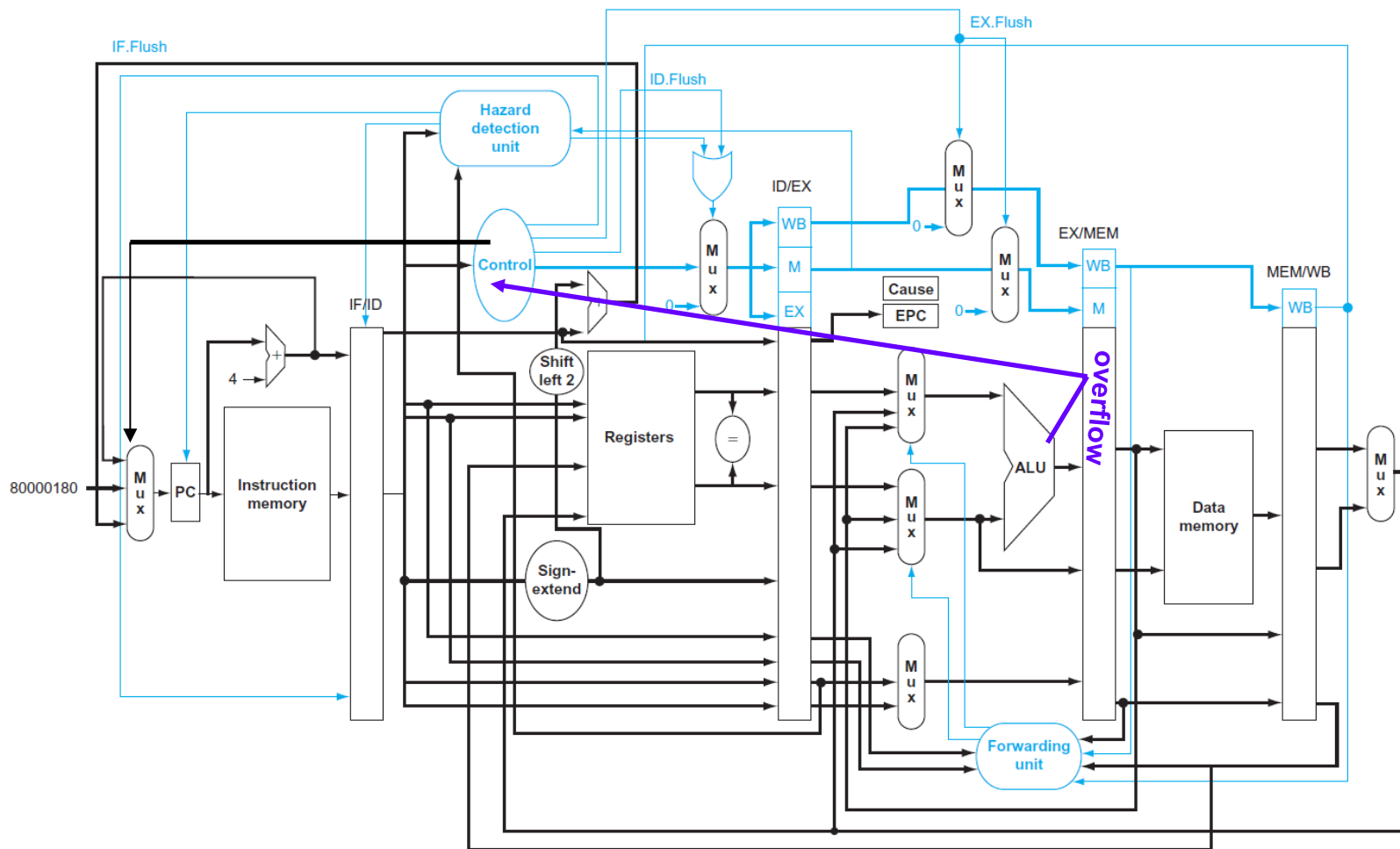
Multiple Simultaneous Exceptions



טיפול במספר חריגות במקביל בצנרת

מכיוון שחריגות יכולות להיווצר במחזורי שעון שונים, לעיתים יכול להיווצר מצב בו שתי חריגות מגיעות בעת ובעונה אחת משתי פקודות מכונה שנמצאות כעת בצנרת. במקרה כזה צריך ליצור מנגנון עדיפויות בין החריגות. המעבד MIPS מעדיף במקרה זה את החריגה שנגרמה על ידי הפקודה שנכנסה מוקדם יותר אל הצנרת, אך קיימים מעבדים אחרים בשוק שמיישמים מנגנוני עדיפות אחרים.

Handling exceptions



גלישה אריתמטית בצנרת

- עיינו בתרשים בעמוד הקודם.
- תרשים זה מראה טיפול בחריגה של גלישה (overflow) בצנרת. אם מתרחשת חריגה כזו, היא תתרחש במחזור השעון השלישי. לכן מוצגים בתרשים זה קווי ה-Flush לאוגרי הצנרת EX/MEM, ID/EX, IF/ID השוטפים את הפקודה שגרמה לגלישה ושאר הפקודות שאחריה בצנרת ומעשה הופכים אותם לבועות הנשטפות בצנרת. שימו לב לאוגר cause המתעדכן בסיבה\מספר החריגה, ולאוגר EPC המקבל את כתובת הפקודה שגרמה לגלישה (למעט במקרה של delay slot).
- ל-PC מוכנס דרך המרבב כתובת הפקודה הראשונה של קטע הקוד של מערכת ההפעלה (0x80000180) המטפל באירוע (exception handler).

EXAMPLE

Exception in a pipelined computer

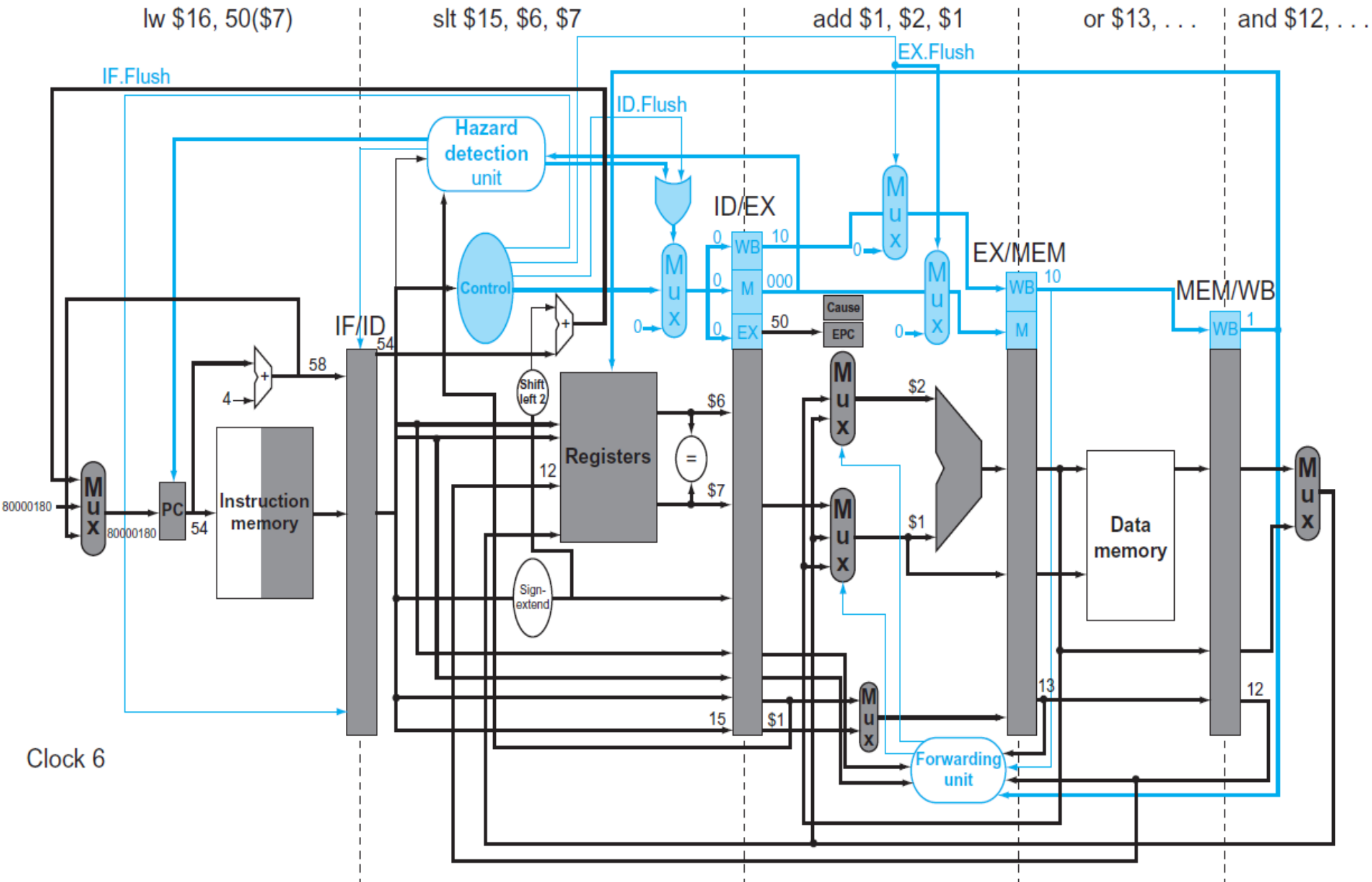
Given this instruction sequence,

```
40hex  sub  $11, $2, $4
44hex  and  $12, $2, $5
48hex  or   $13, $2, $6
4Chex  add  $1,  $2, $1
50hex  slt  $15, $6, $7
54hex  lw   $16, 50($7)
...
```

assume the instructions to be invoked on an exception begin like this:

```
80000180hex  sw   $26, 1000($0)
80000184hex  sw   $27, 1004($0)
...
```

Show what happens in the pipeline if an overflow exception occurs in the add instruction.



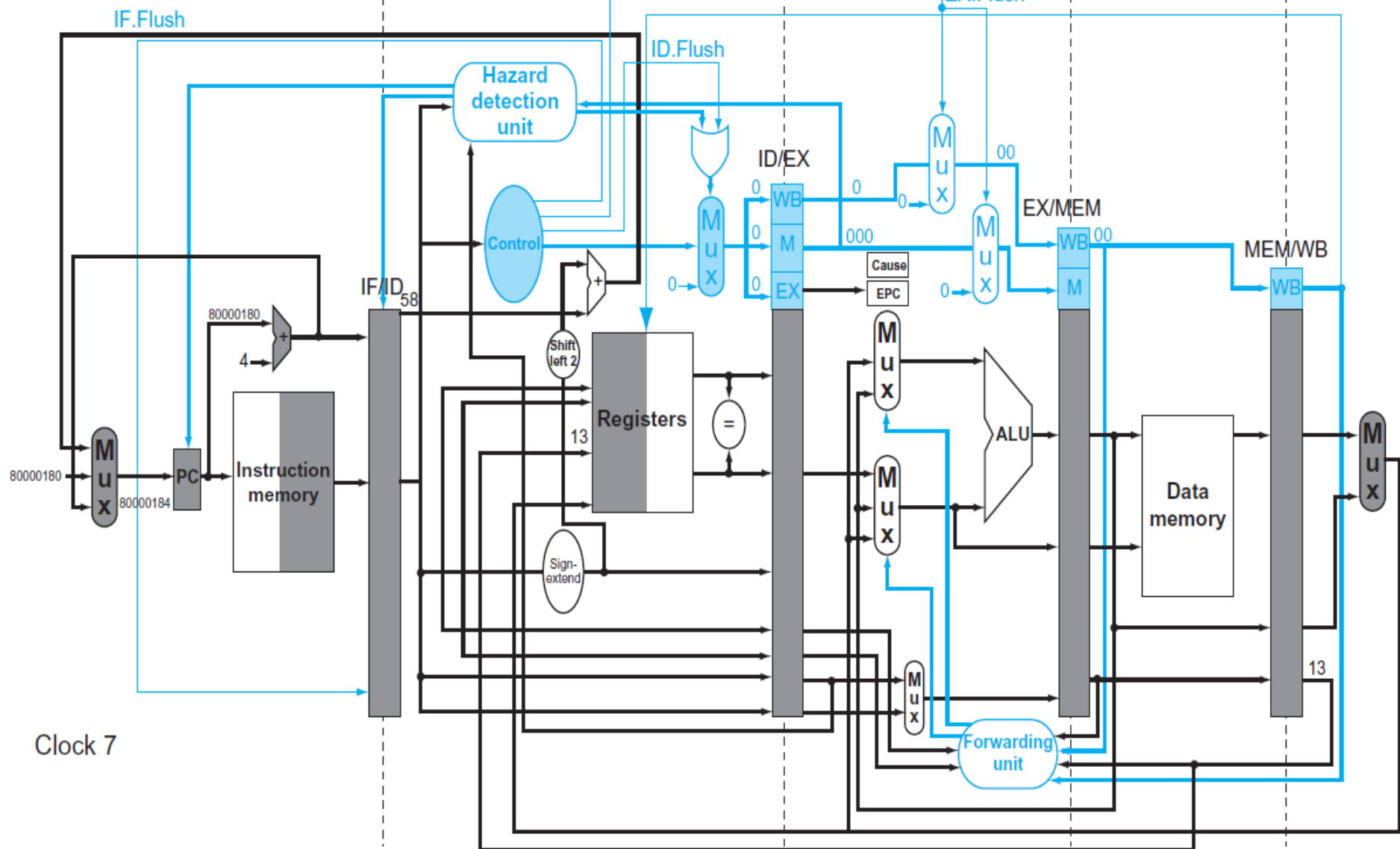
sw \$26, 1000(\$0)

bubble (nop)

bubble

bubble

or \$13, ...



Clock 7

כעת עיינו בתרשימים בשלושת העמודים הקודמים:

- תרשים זה מראה כיצד מתחלף רצף הפקודות לאחר חריגת גלישה בפקודת add. מחזורי השעון 1 עד Clock 5 לא מופיעים בתרשים, אך נסביר גם אותם:
- במחזור השעון 1 הפקודה sub נמצאת במחזור השעון הראשון.
- במחזור השעון 2 הפקודה sub נמצאת במחזור השעון השני והפקודה and נמצאת במחזור השעון הראשון.
- במחזור השעון 3 הפקודה sub נמצאת במחזור השעון השלישי, הפקודה and נמצאת במחזור השעון השני והפקודה or נמצאת במחזור השעון הראשון.
- במחזור השעון 4 הפקודה sub נמצאת במחזור השעון הרביעי, הפקודה and נמצאת במחזור השעון השלישי, הפקודה or נמצאת במחזור השעון השני והפקודה add נמצאת במחזור השעון הראשון.
- במחזור השעון 5 הפקודה sub נמצאת במחזור השעון החמישי, הפקודה and נמצאת במחזור השעון הרביעי, הפקודה or נמצאת במחזור השעון השלישי, הפקודה add נמצאת במחזור השעון השני והפקודה slt נמצאת במחזור השעון הראשון.
- במחזור השעון 6 הפקודה sub כבר לא נמצאת בצנרת כי היא סיימה, הפקודה and נמצאת במחזור השעון החמישי, הפקודה or נמצאת במחזור השעון הרביעי, הפקודה add נמצאת במחזור השעון השלישי וכאן הפקודה add מייצרת חריגת גלישה שגורמת לקפיצה למערכת ההפעלה, אך לתוך הצנרת כבר נכנסו שתי הפקודות הבאות בתור - slt שנמצאת כעת במחזור השעון השני והפקודה lw שנמצאת כעת במחזור השעון הראשון.
- במחזור השעון 7 הפקודה or לא מבוטלת אלא ממשיכה למחזור שעון חמישי כי הפקודה or הייתה לפני הפקודה שגרמה לחריגה, הפקודה add שגרמה לחריגה וכל הפקודות שהתחילו להתבצע אחריה נהפכות ל"בועות". הפקודה sw שהיא הפקודה הראשונה בקוד של מערכת ההפעלה נכנסת לצנרת ומחזור השעון 7 הוא מחזור השעון הראשון שלה.

Additions to MIPS to Handle Exceptions (Fig 4.66)

- Cause register (records exceptions) – hardware to record in Cause the exceptions and a signal to control writes to it (CauseWrite)
- EPC register (records the addresses of the offending instructions) – hardware to record in EPC the address of the offending instruction and a signal to control writes to it (EPCWrite)
 - Exception software must match exception to instruction
- A way to load the PC with the address of the exception handler
 - Expand the PC input mux where the new input is hardwired to the exception handler address - (e.g., 80000180_{hex} for arithmetic overflow)
- A way to flush offending instruction and the ones that follow it



סיימנו...

שאלות?

אפקה

המכללה האקדמית
להנדסה בתל אביב

