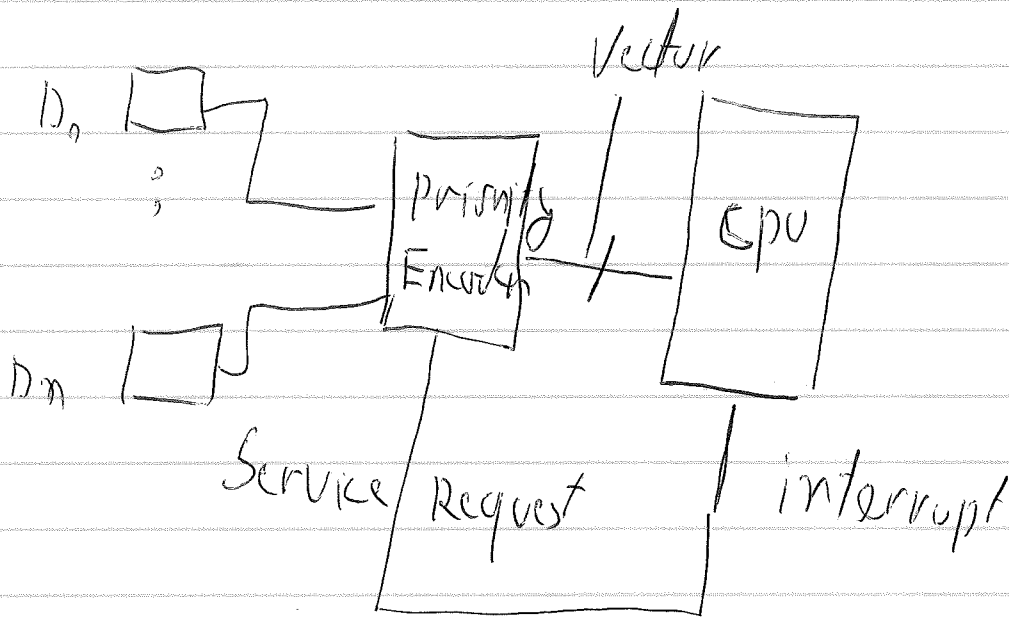


CS2318  
12/11/2016

# Interrupts

Event Driven - Service providing



Vector identifies the device

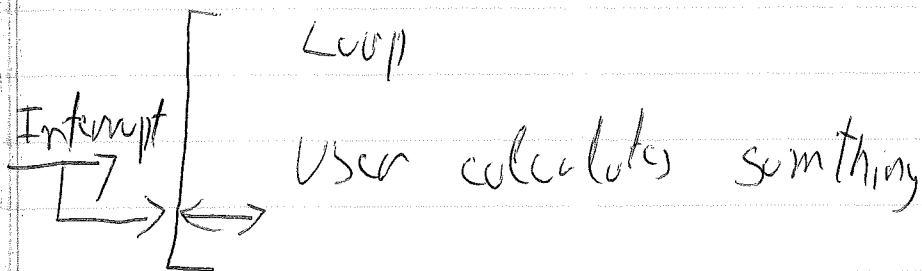
TRAP - has an trap ID

$\Rightarrow$  CPU get an interrupt with ID

$\Rightarrow$  Translates into a call to an  
interrupt service routine

2

Per I/O service routine



Stop processing of User Code

Change flow and execute the specific

Service routine associated with I/O

1000 add \$5, \$6, \$12 ← interrupt  
 → 1004 ← return

$IRA \leftarrow (PC) + 4$  (INT Return Address Register)

$PC \leftarrow \text{Address of routine}$

3

# Interrupt Routine

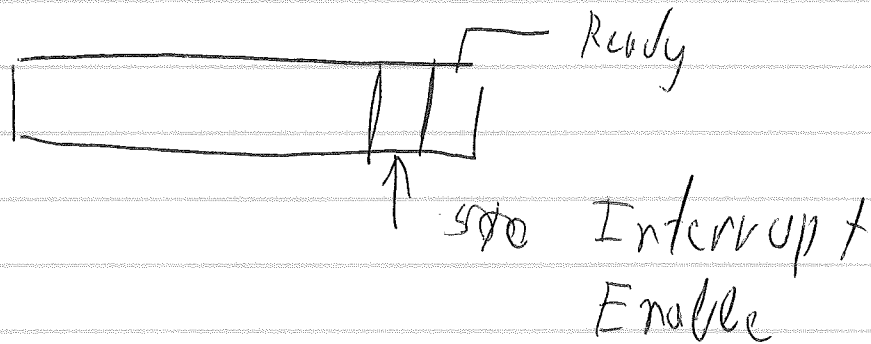
Base 0xFFFF0000  
0004 } KBD

0008  
000C } Screen

FFFF0010 S/C Device

FFFF0014 Data

D0



Polling use the ready

Under Interrupt - no need to check ready

Interrupt Enable ← Changes the

Mode from polling to interrupt

4)

Can be outside the routine

li \$a0, 0xFFFF0000      \$a0 ← base address

Must be outside the routine

~~li \$t0, 0~~    ~~lw \$t0, 16(\$a0)~~

lw \$t0, 0x10(\$a0)     $\Rightarrow$  \$t0 - SIC

ori \$t0, \$t0, 0x0002

0 0 0 0 | 0000 0000 0000 0000  
sw \$t0, 16(\$a0)     $\leftrightarrow$  Enabled interrupts

Alternative li \$t1, 0x00000002

or \$t0, \$t1, \$t0

5

In Routine

DI

Disable interrupts

push some

(push old)

Some - only needed registers

[ Service

pop some

[ old ]

EI

Enable interrupts

IRET

$PC \leftarrow IRA$

Assume the device is an input device

After getting the data it is stored

Pushed to stack Growing Down

SP Points to current data

6

push to

;

lw \$t0, 20(\$a0)

00 - base

subi \$sp, \$sp, 4

tn ← data

of D<sub>0</sub>

sw \$t0, 0(\$sp)

Pop \$t0

EI

IRET

Disable No Interrupts

lw \$t0, 16(\$a0) - SIC is in t<sub>0</sub>

li \$t1, 0xFFFFFFFF1101

And \$t0, \$t0, \$t1

DI - Ignore interrupt

sw \$t0, 16(\$a0)

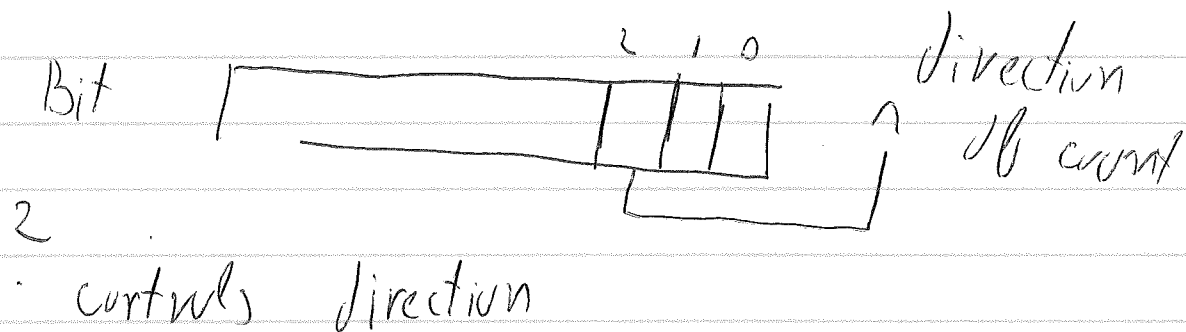
D

7

MIPS AND /ADDI are bitwise

TI	a	b	a AND b	a OR b	a XOR b
	0	0	0	0	0
	0	1	0	1	1
	1	0	0	1	1
	1	1	1	1	0

Assume the device is on up/down counts



Assume we want to flip direction

Find if 1 or 0

8

Find if bit 2 is set

lw \$t0, 16(\$00)

li \$t1, 00000000 0100

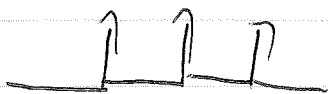
AND \$t0, \$t0, \$t1

beqz \$t0, → depends on the status of the bit.

Device generates 10k interrupts per second

Use This to update a clock

H:M:S

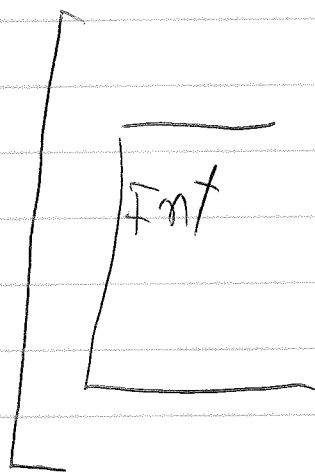


10k per second





9



Assume \$t0 hold h  $\leftrightarrow$

\$t1 || M  $\leftrightarrow$  Mem

\$t2 || S  $\leftrightarrow$

\$t3 second  $\leftrightarrow$

starts at 0 inc per int

when reach 9999 <sup>inc</sup> mod 10k

$\hookrightarrow$  ~~\$t2~~ inc mod 60

$\hookrightarrow$  inc \$t1 mod 60

$\hookrightarrow$  inc \$t0 mod 12/24

10 Computer can count time

can clock compute

Lody A D A Bologe