

Check Your Grades! 1 Week Rule!

Problem Session #7 Cancelled This Week!

Free Points!

HW #8 Due 10/29

HW #9 Due 11/5

<https://powcoder.com>

Problem Session #8 (Last) Week of 11/2

Program #3 Due 11/11/20 Assignment Project Exam Help

Late Assignments -10% per day!

[Assignment Project Exam Help](https://powcoder.com)

Today:

Chapter 8 Input/Output <https://powcoder.com>

Chapter 9 Subroutines

[Add WeChat powcoder](https://powcoder.com)

TopHat IC9

From Syllabus

Regrading Requests

If you have discussed your grading on an assignment with your TA and are still not satisfied, you may submit a request to me within **one (1) week** of the graded assignment being returned to you. You must write a cover sheet explaining why you feel you deserve additional points on a given problem, attach it to the front of your graded paper, and give it to me either in class or my office. Regrading requests will **NOT** be considered more than one week after the assignments are returned to the students.

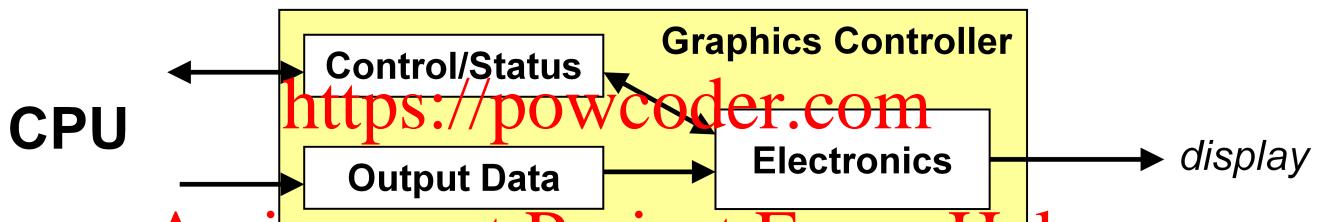
I/O Controller

Control/Status Registers

- CPU tells device what to do -- write to control register
- CPU checks whether task is done -- read status register

Data Registers

- CPU transfers data to/from device



Device electronics

- performs actual operations
 - pixels to screen, bits to/from disk, characters from keyboard

<https://powcoder.com>

Add WeChat powcoder

Transfer Control

Who determines when the next data transfer occurs?

Polling

- CPU keeps checking status register until new data arrives OR device ready for next data
- “Are we there yet? Are we there yet? Are we there yet?”

<https://powcoder.com>

Interrupts

- Device sends a special signal to CPU when new data arrives OR device ready for next data
- CPU can be performing other tasks instead of polling device.
- “Wake me when we get there.”

<https://powcoder.com>

Add WeChat powcoder

LC-3

Memory-mapped I/O (Table A.3)

Location	I/O Register	Function
xFE00	Keyboard Status Reg (KBSR)	Bit [15] is one when keyboard has received a new character.
xFE02	Keyboard Data Reg (KBDR)	Bits [7:0] contain the last character typed on keyboard.
xFE04	Display Status Register (DSR)	Bit [15] is one when device ready to display another char on screen.
xFE06	Display Data Register (DDR)	Character written to bits [7:0] will be displayed on screen.

Assignment Project Exam Help

Asynchronous devices

- synchronized through status registers

Assignment Project Exam Help

Polling and Interrupts

- the details of interrupts will be discussed in Chapter 10

Assignment Project Exam Help

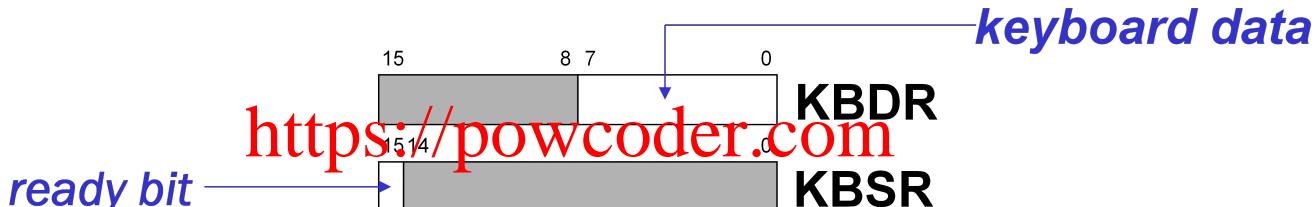
Assignment Project Exam Help

Add WeChat powcoder

Input from Keyboard

When a character is typed:

- its ASCII code is placed in bits [7:0] of KBDR (bits [15:8] are always zero)
- the “ready bit” (KBSR[15]) is set to one
- keyboard is disabled -- any typed characters will be ignored



Assignment Project Exam Help

When KBDR is read:

- KBSR[15] is set to zero
- keyboard is enabled

https://powcoder.com

Keyboard Echo Routine

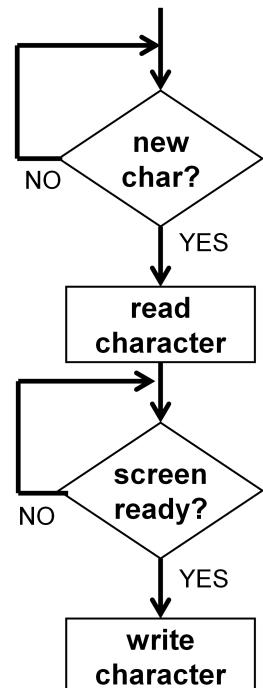
Add WeChat powcoder

- User gets feedback on character typed and knows its ok to type the next character.

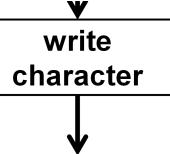
ETC

On T

```
POLL1    LDI   R0 , KBSRPtr
          BRzp POLL1
          LDI   R0 , KBDRPtr
          STI   R0 , DDRPtr
...
KBSRPtr .FILL xFE00
KBDRPtr .FILL xFE02
DSRPtr  .FILL xFE04
DDRPtr  .FILL xFE06
```



```
DSRPtr .FILL xFE04  
DDRPtr .FILL xFE06
```



Polling Overhead

Options for polling

- Don't always need to sit in tight polling loop, doing nothing else.
- But must poll device often enough to not miss any potential data transfers.

Assignment Project Exam Help

Example: mouse

- Assignment Project Exam Help
- Must poll mouse at least 30 times per second, to see if (a) change of position and/or (b) button pressed/released.
 - How much overhead does this represent?
 - In other words, what percentage of CPU cycles will be used up checking for mouse events?

Mouse Overhead -- Polling

Assume the following parameters:

- CPU clock is 1 GHz (1×10^9 cycles/sec)
- time to service device is 400 cycles, including:
 - transfer to polling routine, checking status, reading data, updating cursor, returning to user code

How much overhead?

<https://powcoder.com>

- time spent polling = $30 \times 400 = 12,000$ cycles/sec
- fraction of CPU cycles = $12 \times 10^3 / 1 \times 10^9 = 0.00012\%$

[Assignment Project Exam Help](https://powcoder.com)

- Conclusion: Polling is not a lot of overhead in this case.
- Not counted -- how do we count when to do? via interrupt?

Network Overhead -- Polling

What about a network device?

Add WeChat powcoder 3 20b

- 100 Mb/s Ethernet, each "packet" is 40B
- Have to poll often enough to not drop data.
- 1 GHz processor, 400 cycles to poll + read data

How much overhead?

- $(320 \text{ bits} / 10^8 \text{ bits/sec}) \times (10^9 \text{ cycles/sec})$
= 1 packet every 3200 cycles = 312,500 packets/sec
- fraction of CPU cycles
= $(400 * 312,500 \text{ cycles/sec}) / (10^9 \text{ cycles/sec}) = 12.5\%$
- This seems a little much -- will interrupts help?

Interrupt-Driven I/O

External device can:

- (1) Force currently executing program to stop;
- (2) Have the processor satisfy the device's needs; and
- (3) Resume the stopped program as if nothing happened.

Why?

<https://powcoder.com>

- Polling consumes a lot of cycles, especially for rare events – these cycles can be used for more computation.
- Example: Process previous input while collecting current input. (See Example 8.1 in text - page 221.)

[Assignment Project Exam Help](#)

[Assignment Project Exam Help](#)

Network Overhead - Interrupts

Why interrupts?

[Add WeChat powcoder](#)

- Network is not "active" 100% of the time.
Many times, we'll poll and find out there's no data to get.
- Interrupt will happen ONLY when there's actually a packet to be transferred.
- Let's assume network is active 10% of the time.



Overhead?

- $312,500 \text{ packets/sec} \times 10\% = 31,250 \text{ packets/sec}$
- fraction of CPU cycles
 $= (400 \times 31,250 \text{ cycles/sec}) / (10^9 \text{ cycles/sec}) = 1.25\%$

Priority

Every instruction executes at a stated level of urgency.

LC-3: 8 priority levels (PL0-PL7)

- Example:
 - Payroll program runs at PL0.
 - Nuclear power correction program runs at PL 6.
- It's OK for PL6 device to interrupt PL0 program,
<https://powcoder.com>
but not the other way around.

Assignment Project Exam Help
Priority encoder selects highest-priority device,
compares to current processor priority level,
and generates interrupt signal if appropriate.

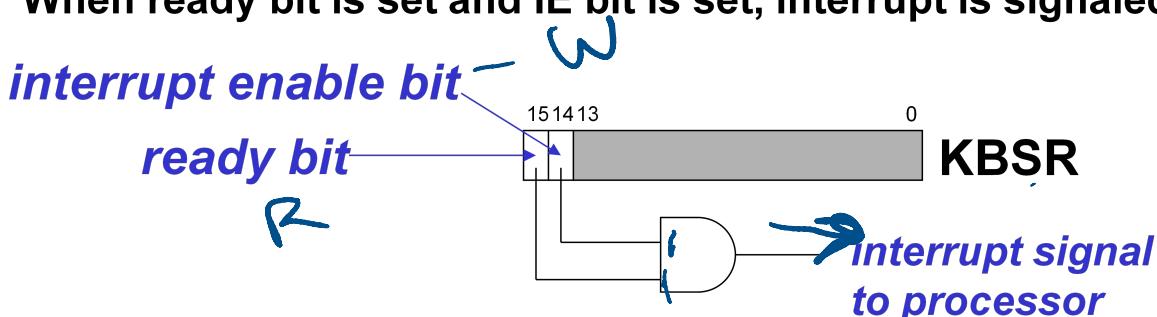
Interrupt-Driven I/O

To implement an interrupt mechanism, we need:

- A way for the I/O device to **signal** the CPU that an interesting event has occurred.
- A way for the CPU to **test** whether the **interrupt signal** is set and whether its **priority is higher** than the current program.

Generating Signal

- Software sets "interrupt enable" bit in device register.
- When ready bit is set and IE bit is set, interrupt is signaled.

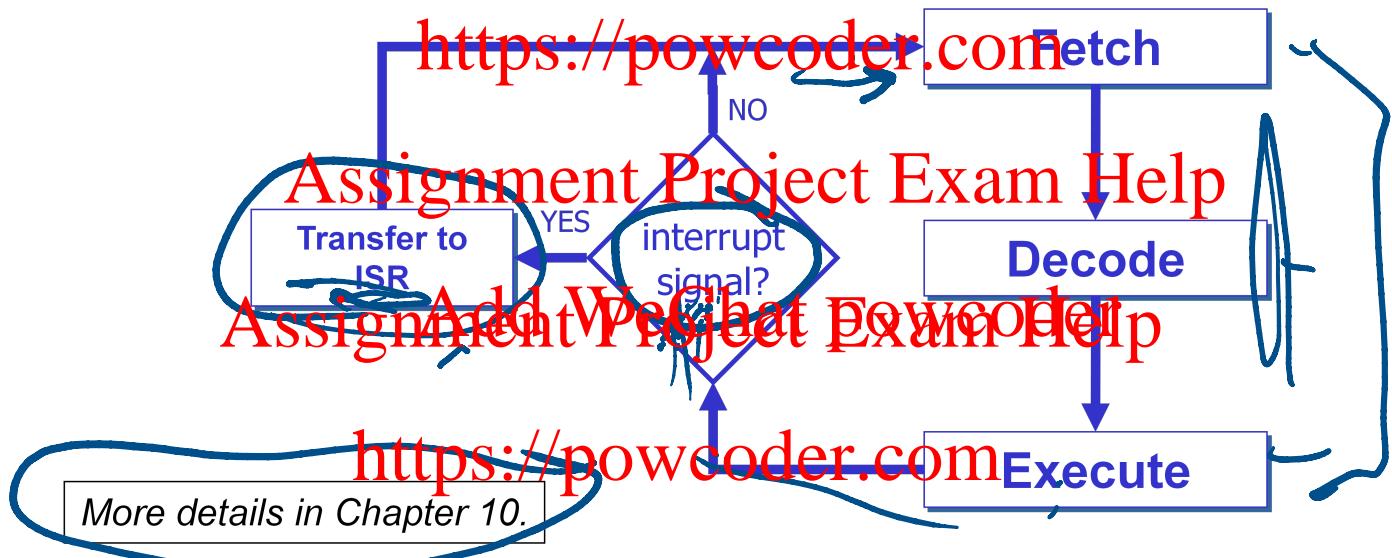


Testing for Interrupt Signal

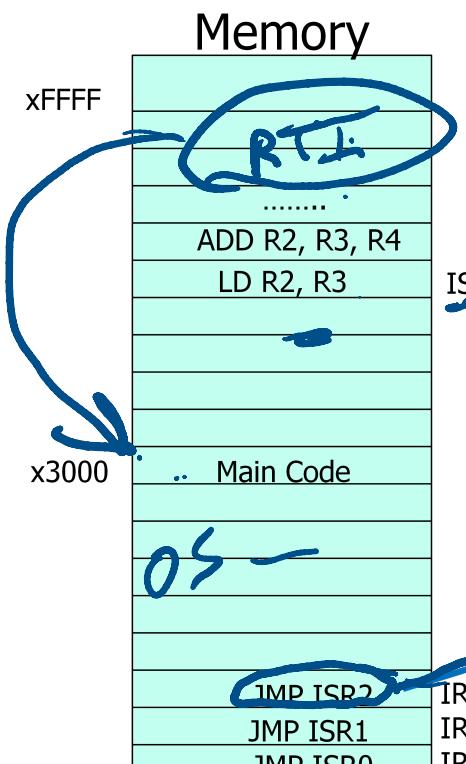
CPU looks at signal between EXECUTE and FETCH phases.

If not set, continues with next instruction.

If set, transfers control to interrupt service routine.



Interrupt Service Routines (ISR) - small bit of code + do interrupt + & tasks.





<https://powcoder.com>

Assignment Project Exam Help

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder