
UCCD1133

Introduction to Computer Organisation and Architecture

Assignment Project Exam Help

<https://powcoder.com>

Chapter 7

Add WeChat powcoder

Computer Peripherals

Disclaimer

- This slide may contain copyrighted material of which has not been specifically authorized by the copyright owner. The use of copyrighted materials are solely for educational purpose. If you wish to use this copyrighted material for other purposes, you must first obtain permission from the original copyright owner.

Assignment Project Exam Help

<https://powcoder.com>

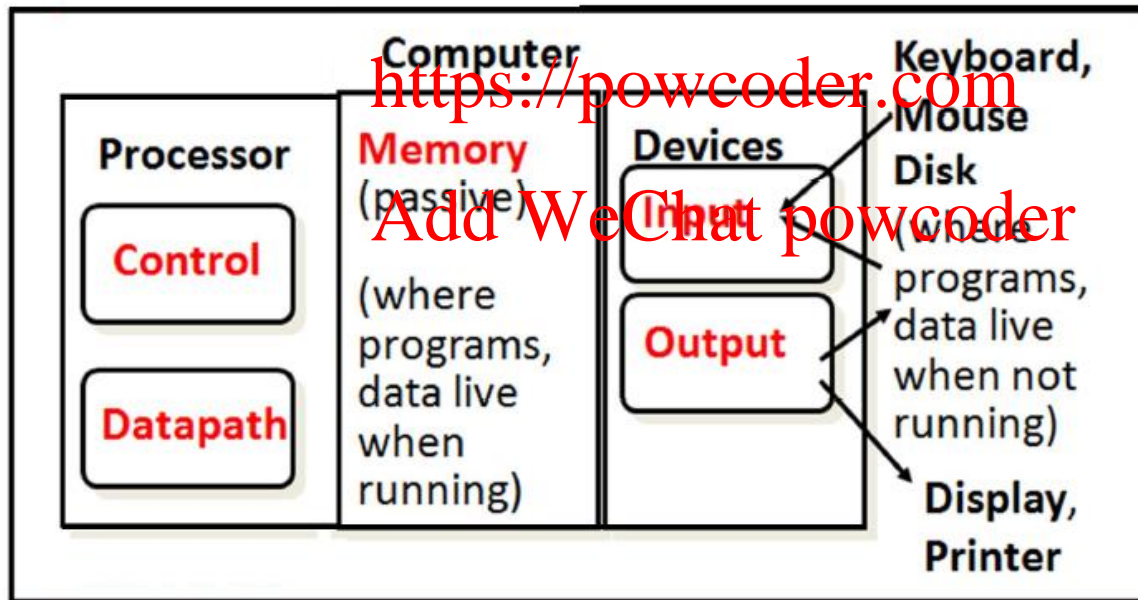
Add WeChat powcoder

Computer I/O

- I/O is one of the most complex areas of computer design
 - Wide variety of different I/O devices
 - The quest for fast handling of programs and data

Five Major Components of Computer Organization

Assignment Project Exam Help



CD-ROM,
modems,
scanners,
sound cards
with speakers
and
microphones,
...

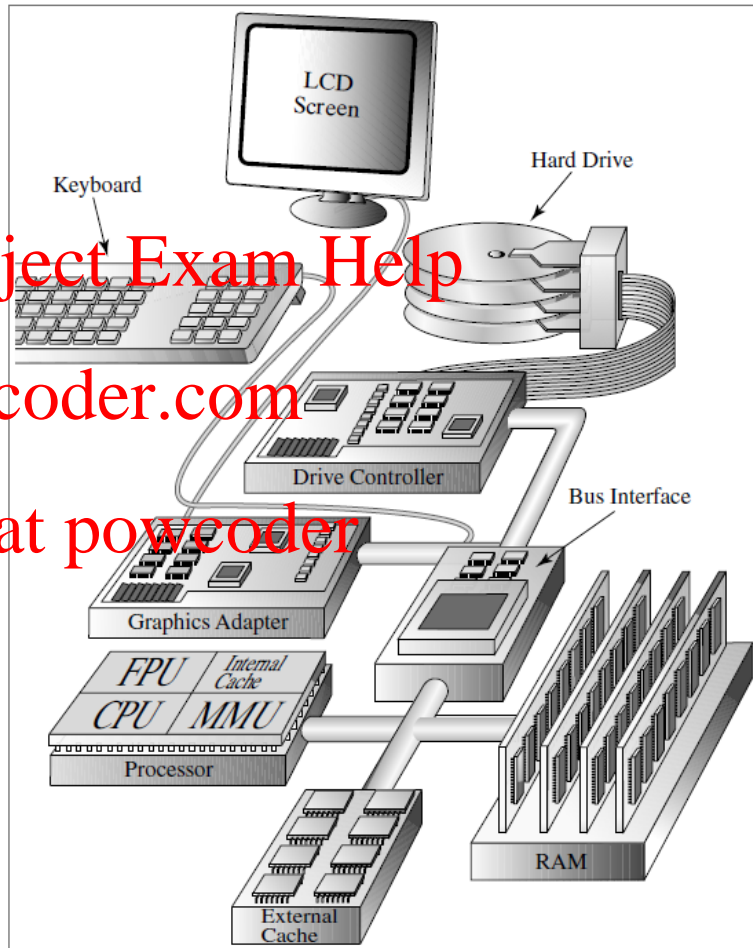
Peripherals

- I/O devices attached to the computer online are called peripherals.
 - Connected online = control by CPU.
- Example: Keyboard, hard drive, graphic display etc.

Assignment Project Exam Help

<https://powcoder.com>

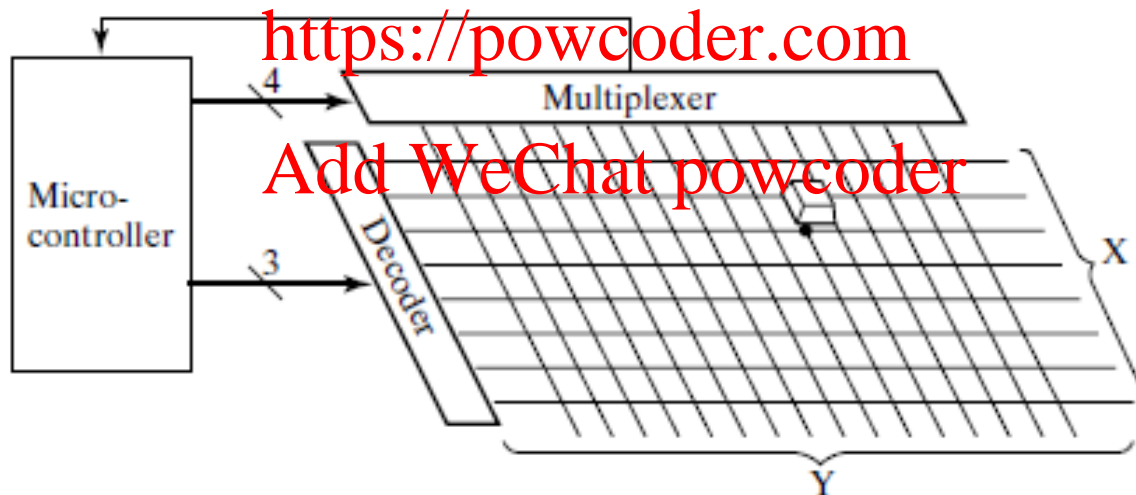
Add WeChat powcoder



Example: Keyboard

- One of the slowest data rates.
- It is necessary to detect which of the keys have been depressed/ released.
 - A scan matrix lies beneath the keys.
 - Microcontroller is programmed to periodically scan all intersections in the matrix.
- Data transfer rate less than 10 bytes/s.

Assignment Project Exam Help



Keyboard Scan Matrix

I/O Interfaces

- The properties of peripherals and CPU & memory are different:

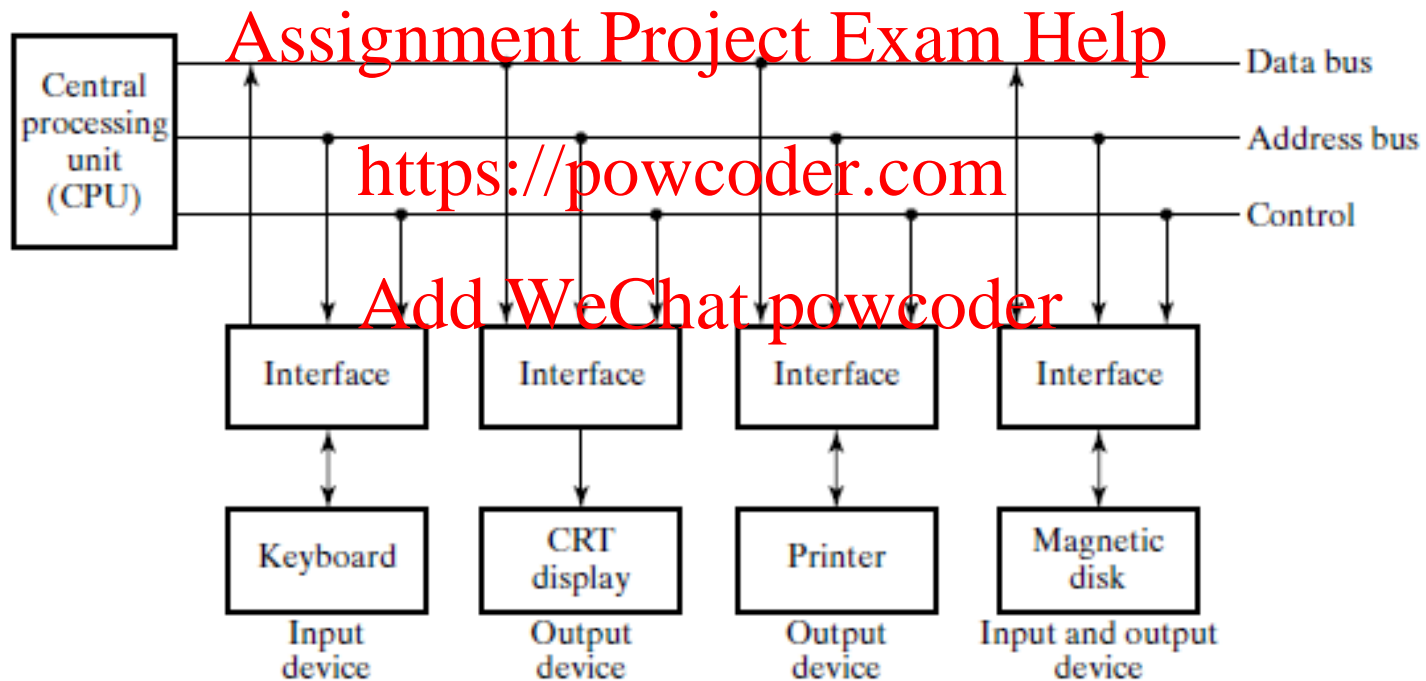
Peripherals	CPU & Memory
Electromechanical devices	Electronic devices
Usually slow data transfer rate	Usually faster than peripherals
Data unit = byte, block, ...	Data in word format

<https://powcoder.com>

- The computer need special hardware to interface the peripherals with CPU: **Interface units**
 - Interface between the **bus** from the **CPU** and **peripheral devices**.
 - Supervise and synchronize all input and output transfer.
 - Each device has its own controller to supervise the operations of the particular mechanism of that peripherals
 - Example: the controller in a printer controls the motion of paper, timing of printing, the selection of the characters to be printed.

I/O Bus and Interface Unit

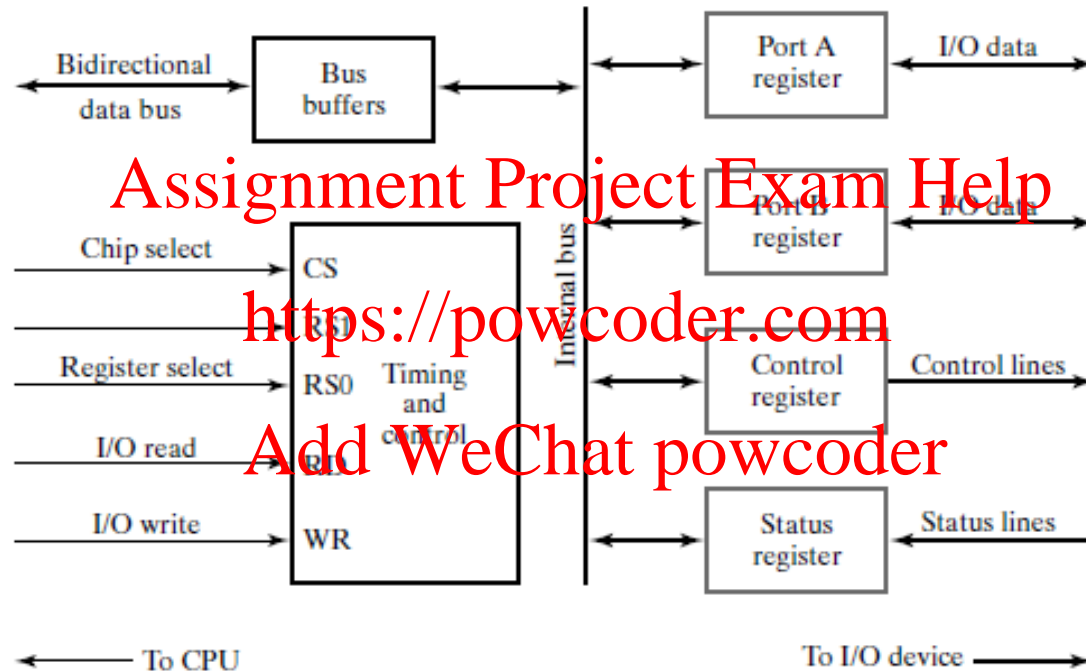
- Bus = a group of electrical signals / wires that carry these signals.
- The **three major components** of system bus:
 - **Data bus:** Carry data from source to destination (e.g., CPU to memory)
 - **Address bus:** Used to designate source or destination of the data.
 - **Control bus:** Carry functional information defining and controlling the current bus activity, e.g., clock, memory R/W, IO R/W, Bus request, etc.



Connection of I/O devices to CPU

Interface Unit

- Function of the interface:
 - Translate the signals between the CPU buses and the I/O devices.
 - Provide the needed hardware to satisfy the two sets of timing constraints.



CS	RS1	RS0	Register selected
0	X	X	None: data bus in high-impedance state
1	0	0	Port A register
1	0	1	Port B register
1	1	0	Control register
1	1	1	Status register

Example of I/O Interface Unit

I/O Data Transfer

- I/O requests are made by applications or OS
 - involve moving data between peripheral device and main memory
- Two main ways for programs to communicate with devices:
 - Memory-mapped I/O
 - Isolated I/O
- Several ways to transfer data between devices and main memory:
 - Programmed I/O
 - Interrupt-driven I/O
 - Direct memory access

Assignment Project Exam Help

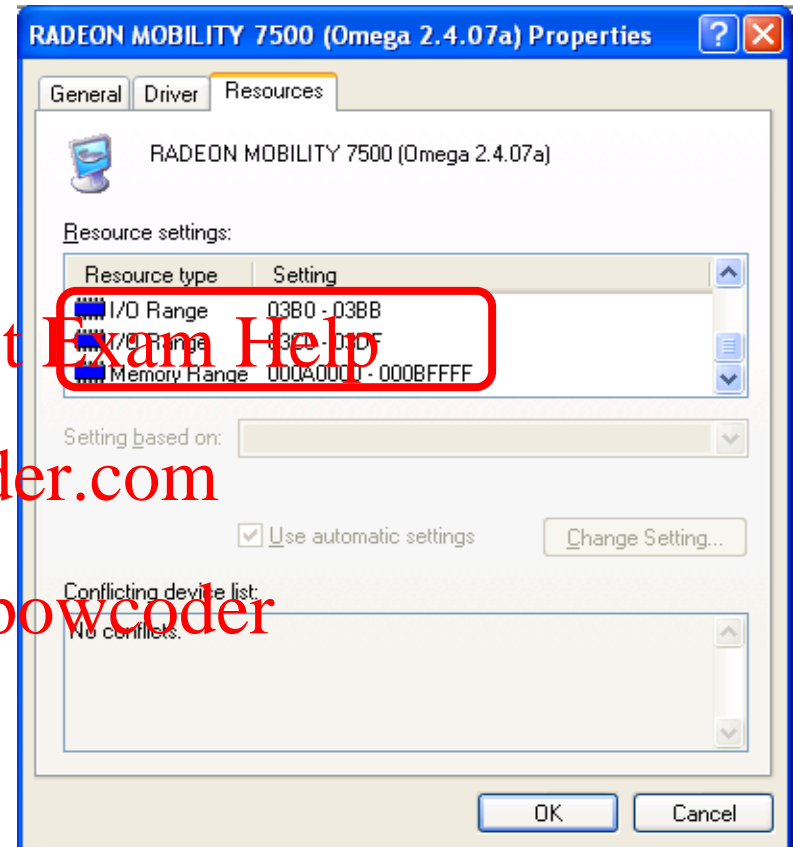
<https://powcoder.com>

Add WeChat powcoder

Accessing I/O Devices

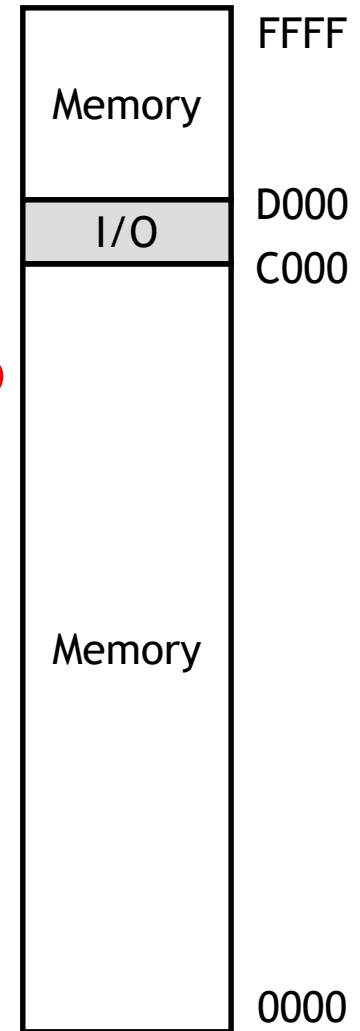
- Most devices can be considered as memories, with an “address” for reading or writing
- Many ISAs* often make this analogy explicit — to transfer data to/from a particular device, the CPU can access special addresses
 - *Example:* Video card can be accessed via addresses 3B0-3BB, 3C0-3DF and A0000-BFFFF
- Two ways to access these addresses:
 - Memory-mapped I/O
 - Isolated I/O

*ISA (Industry Standard Architecture) is a type of computer bus and expansion card standard.



Memory-mapped I/O

- With **memory-mapped I/O**, one address space is divided into two parts.
 - Some addresses refer to physical memory locations.
 - Other addresses actually reference peripherals.
- For example, an old PC with a 16-bit address bus could access a whole 64 KB of memory.
 - Addresses **C000-CFFF** in hexadecimal were not part of memory, but were used to access I/O devices.
 - All the other addresses did reference main memory.
- The I/O addresses are shared by many peripherals. For instance, C010 might be attached to the keyboard while C030 goes to the speaker.
- Some devices may need several I/O addresses.



Memory-mapped I/O

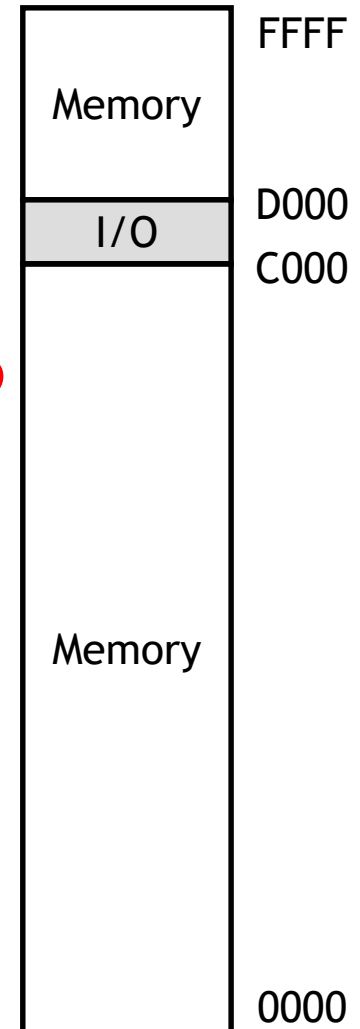
- In summary, memory-mapped I/O maps I/O port address to memory address space.
 - Writing to an I/O port is similar to writing to a memory location.
 - Does not require any special consideration from the processor.
 - However, it takes part of the memory address space for I/O mapping.

Assignment Project Exam Help

<https://powcoder.com>

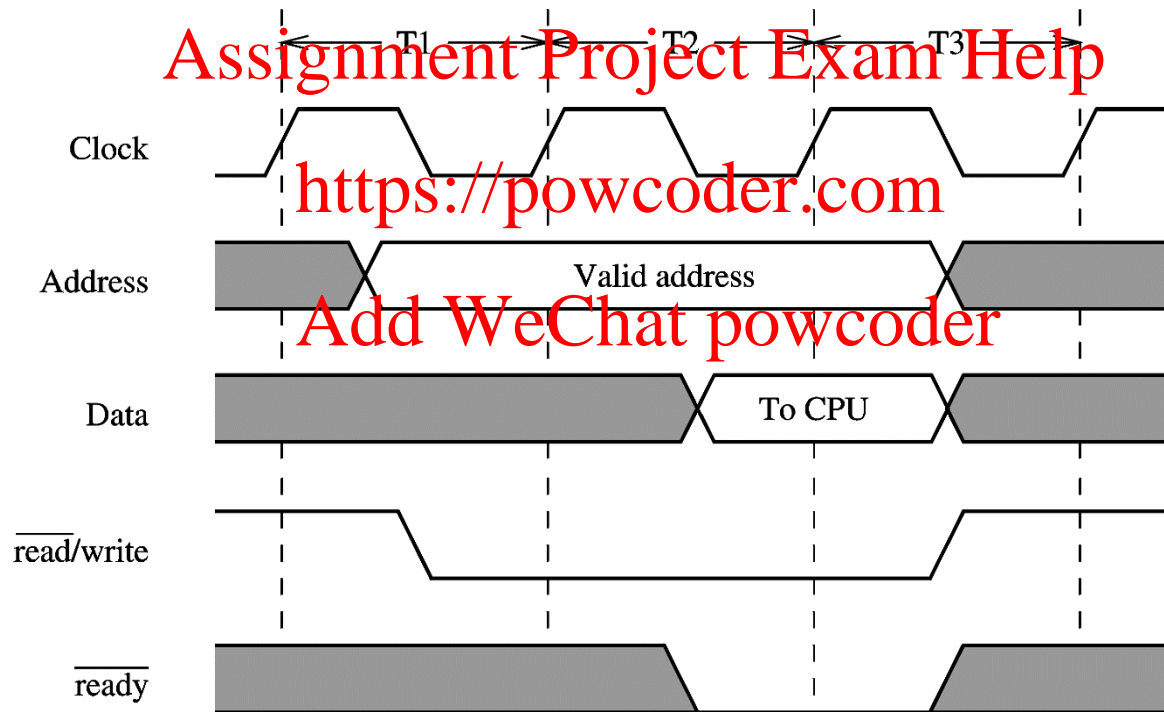
- Processors such as MIPS and PowerPC support only memory-mapped I/O.

Add WeChat powcoder



Memory-mapped I/O

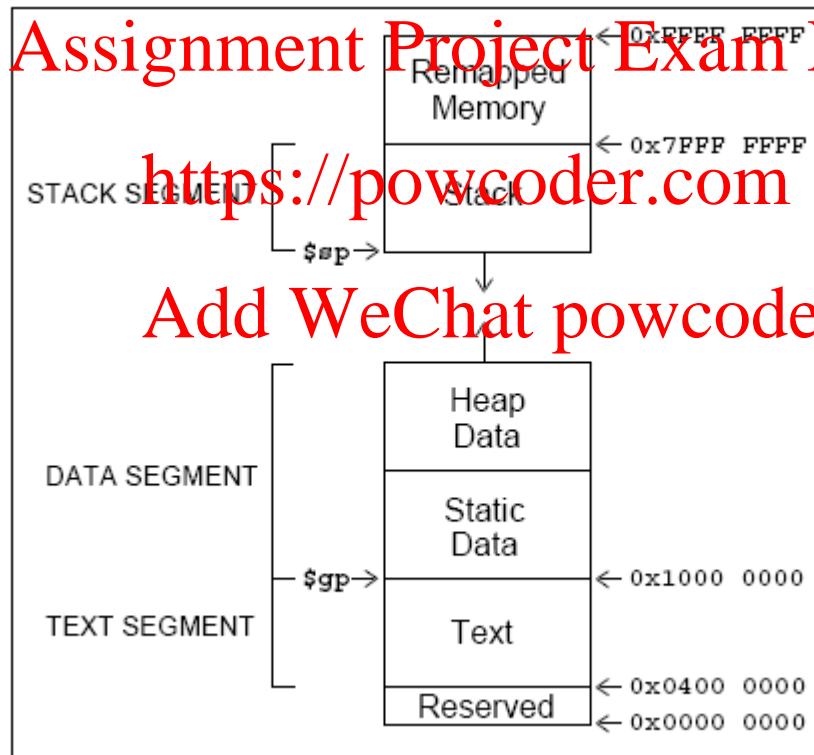
- To send a data to a device,
 - The CPU places a **device address** on the **address bus** to activates the path.
 - The CPU provides a **function code** on the **control lines**.
 - The interface proceed to execute the function and synchronise the **data transfer**.



Synchronous bus (Changes occur relative to the falling or rising edge of the clock)

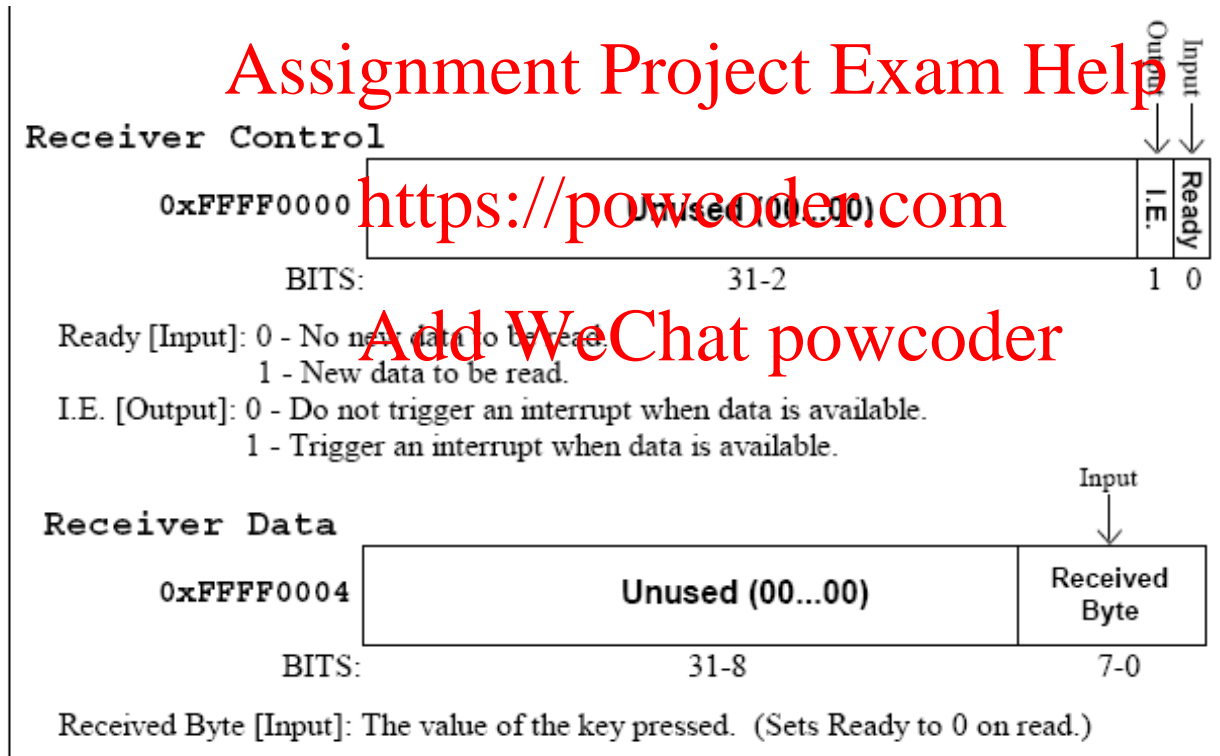
MIPS Memory-mapped I/O

- MIPS processor uses memory-mapped I/O.
- Everything above 0x7FFF FFFF is used by the system
 - Special operating system functions
 - I/O registers mapped to memory address
 - Kernel data, etc.



SPIM Input

- ❑ SPIM allows you to read from the keyboard (which is similar to reading something from the true I/O register)
- ❑ Input is controlled via two memory-mapped device registers:
 - ❑ Receiver control
 - ❑ Receiver data



SPIM output

- Transmitter writes characters to the display (console).
- Output is controlled via two memory-mapped device registers:
 - Transmitter control (0xFFFF 0008)
 - Bit 1: interrupt enable
 - Bit 0: ready
 - Transmitter data (0xFFFF 000C)
 - Bit 0-7: data byte
- If you need to show something on the console, do the following:
 1. Check if ready bit is 1. If yes, proceed. Otherwise, wait.
 2. Write to the data. The ready bit will be reset to 0, and will be set to 1 after the byte is transmitted.

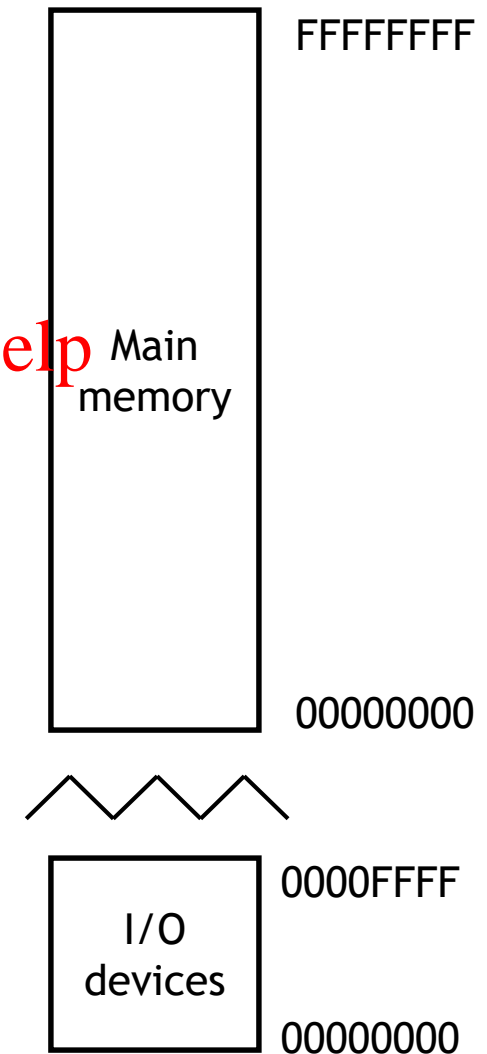
Assignment Project Exam Help

<https://powcoder.com>

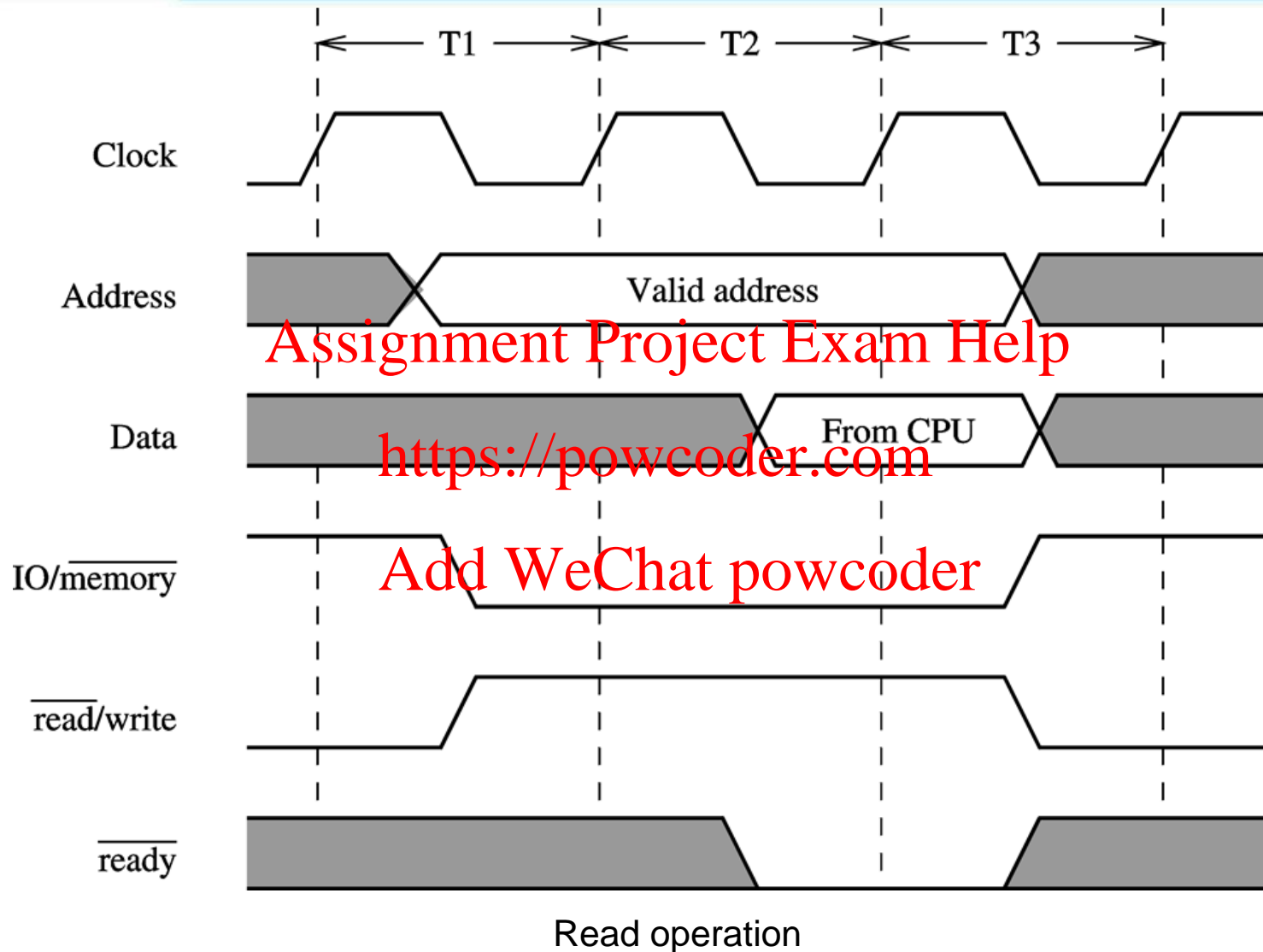
Add WeChat powcoder

Isolated I/O

- For **isolated I/O**, there are *separate* address spaces for memory and I/O devices
 - special instructions that access the I/O space
- The complete memory address space can be used for memory, but it requires special I/O instructions to access the I/O address space.
- Example (x86):*
 - regular instructions like `MOV` reference RAM
 - special instructions `IN` and `OUT` access a separate I/O address space
- An address could refer to *either* main memory or an I/O device, depending on the instruction used.



Isolated I/O



Modes of Data Transfer

- How data is transferred between peripherals and main memory?
- Several modes:
 - Programmed I/O
 - Interrupt-driven I/O
 - Direct memory access

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Programmed I/O

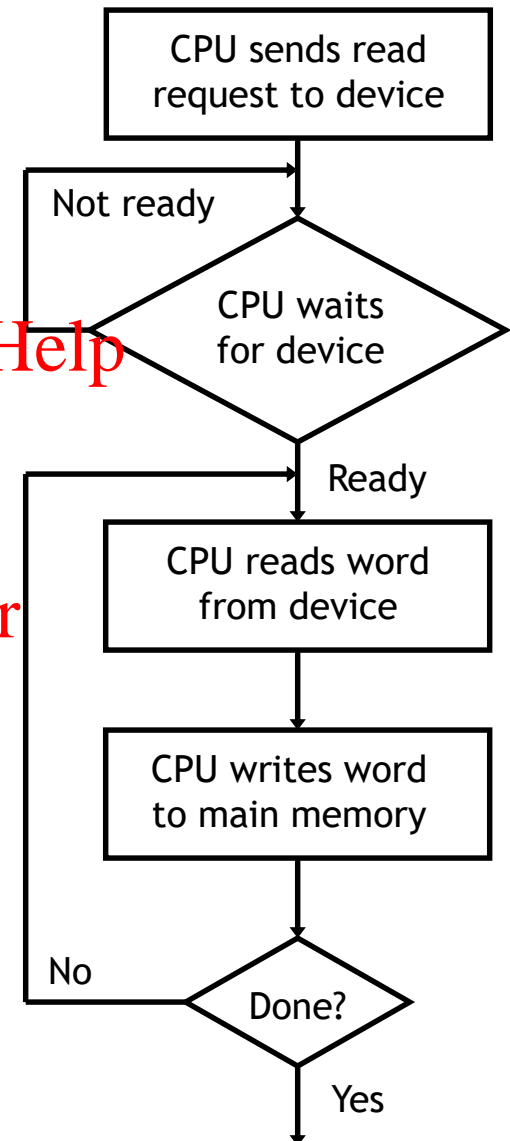
- With **programmed I/O**, it's all up to the program or the OS
 - CPU makes a request and then waits (loops) until device is ready (loop 1)
 - Buses are typically 32-64 bits wide, so loop 2 is repeated for large transfers

Assignment Project Exam Help

- A lot of CPU time is needed for this!
 - most devices *are* slow compared to CPUs
 - CPU also “wastes time” doing actual data transfer

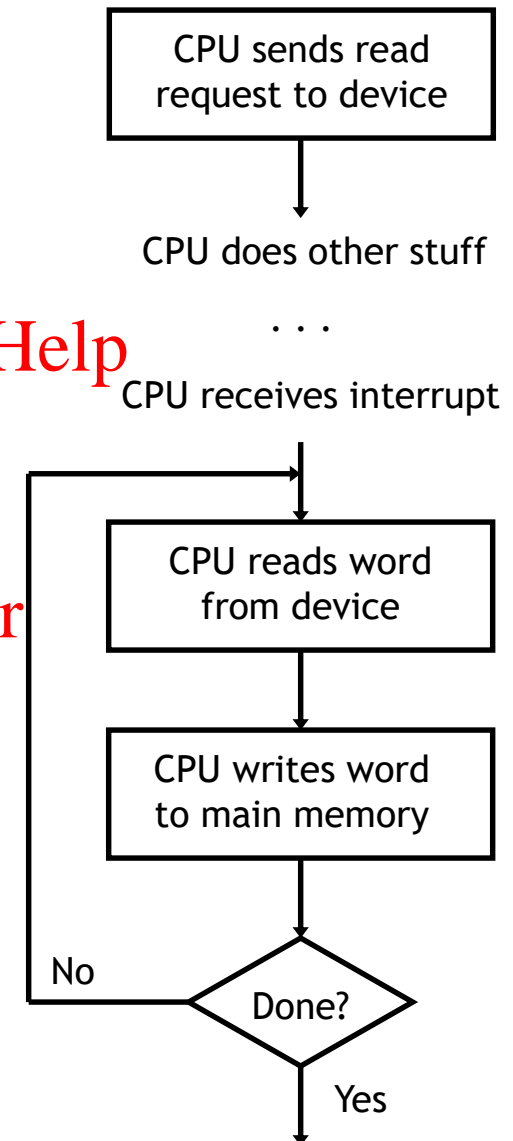
Add WeChat powcoder

- Does not require any hardware support to implement.

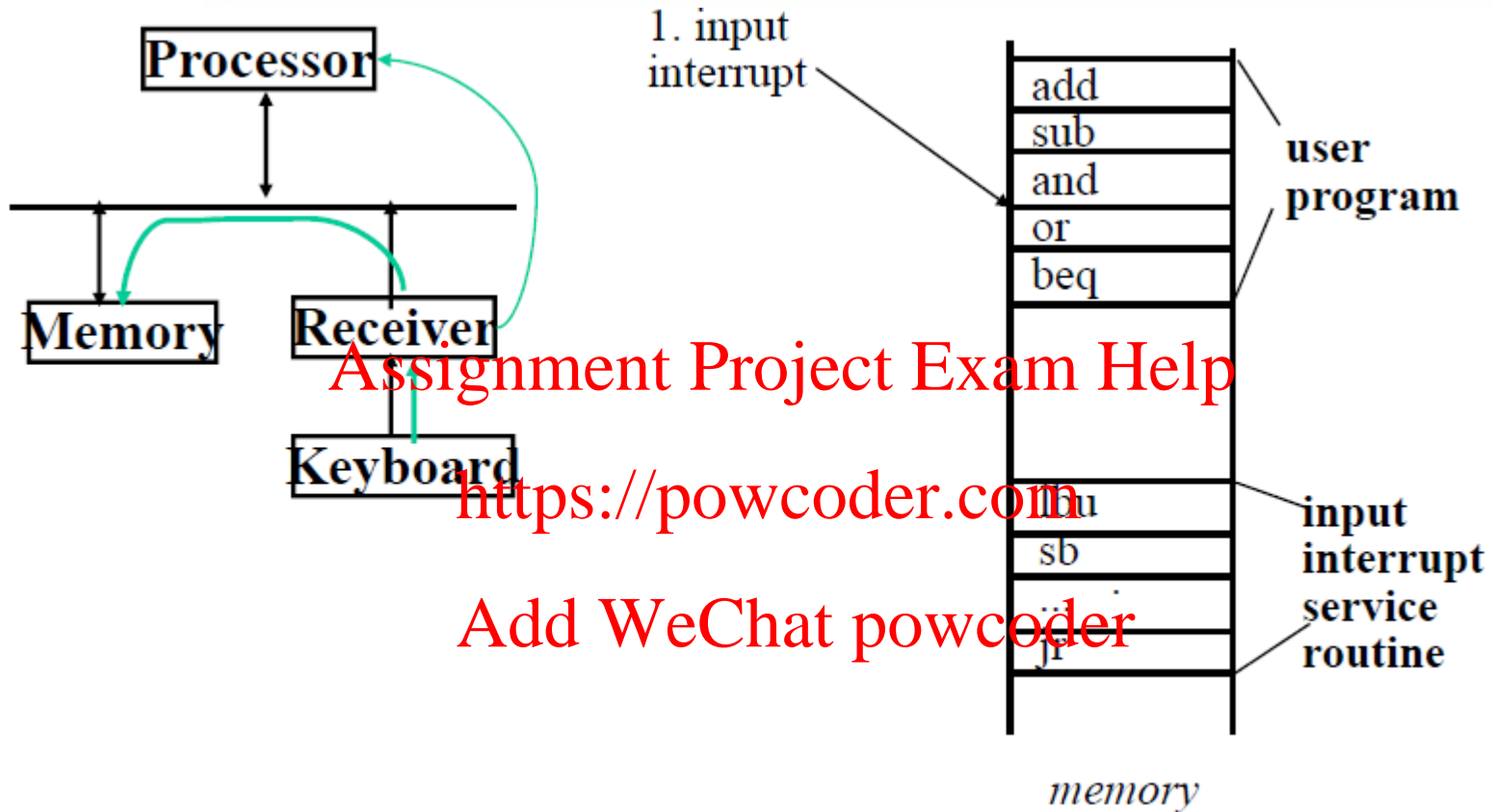


Interrupt-driven I/O

- Continually checking to see if a device is ready is called **polling**
- Wastes CPU time:
 - CPU must ask repeatedly
 - CPU must ask often enough to ensure that it doesn't miss anything, which means it can't do much else while waiting
- Solution: **Interrupt-driven I/O**
 - Instead of waiting, the CPU continues with other calculations
 - The device **interrupts** the processor when the data is ready
- CPU still does the data transfer



Interrupt Driven Input

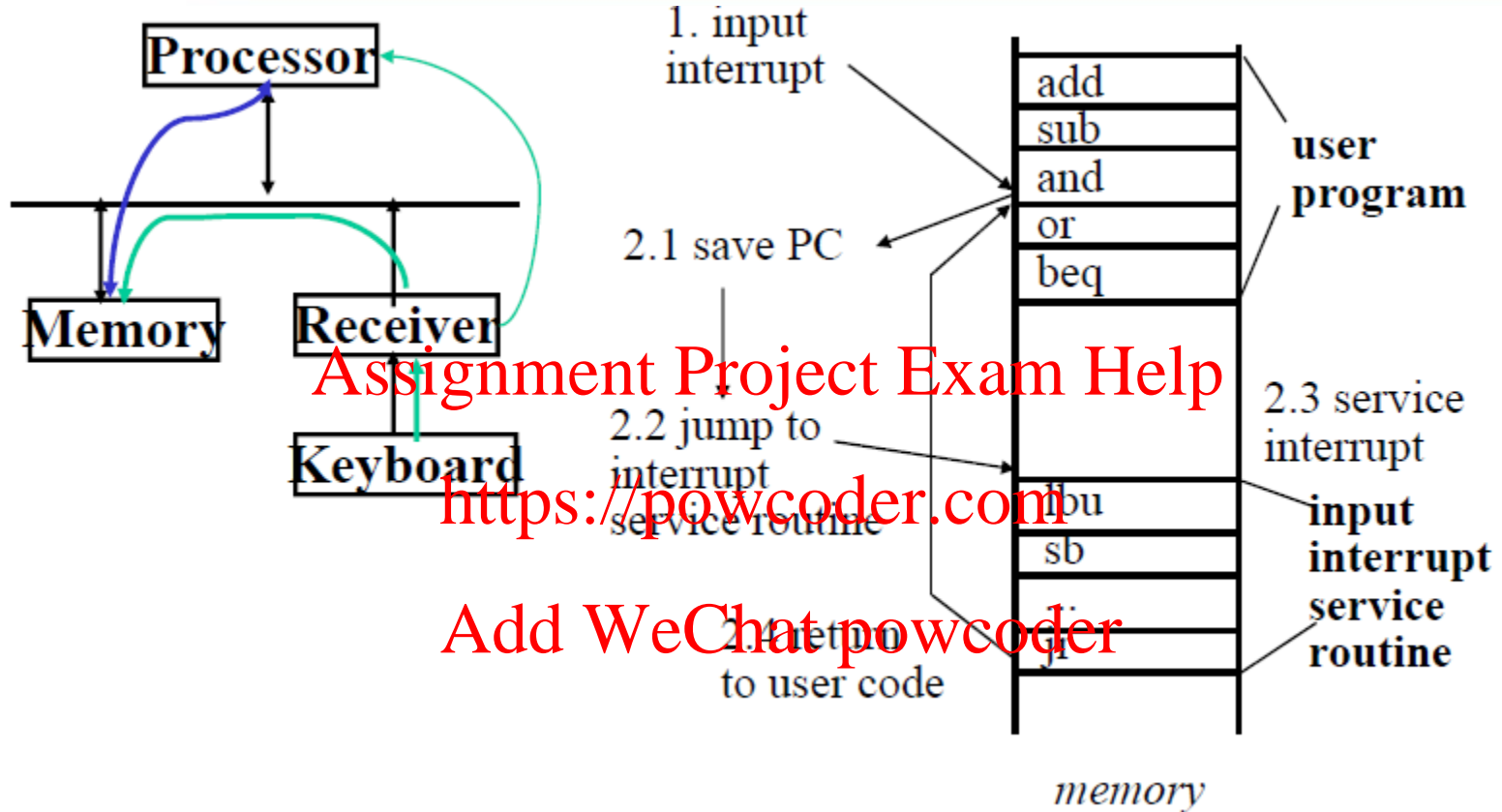


Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Interrupt Driven Input

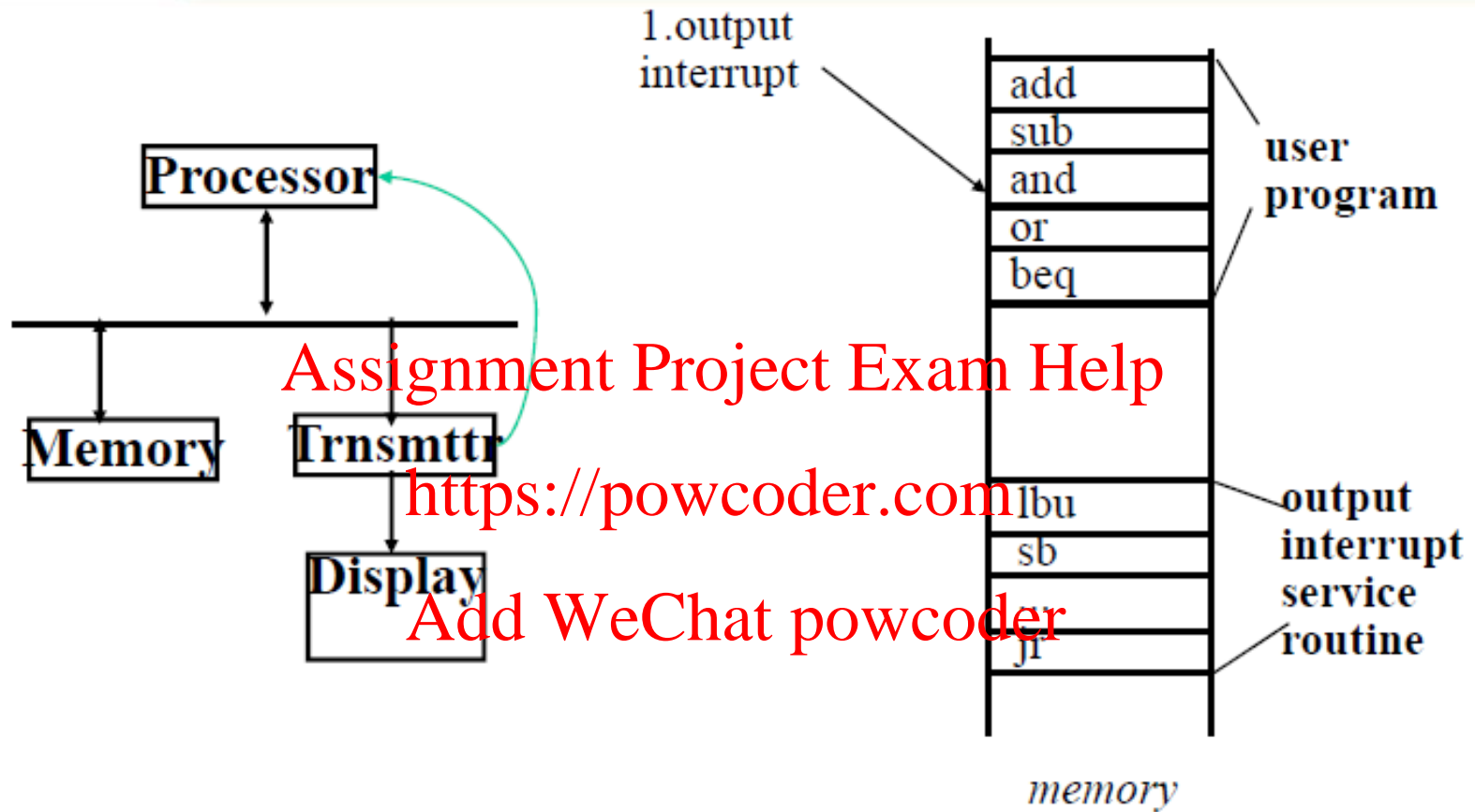


Assignment Project Exam Help

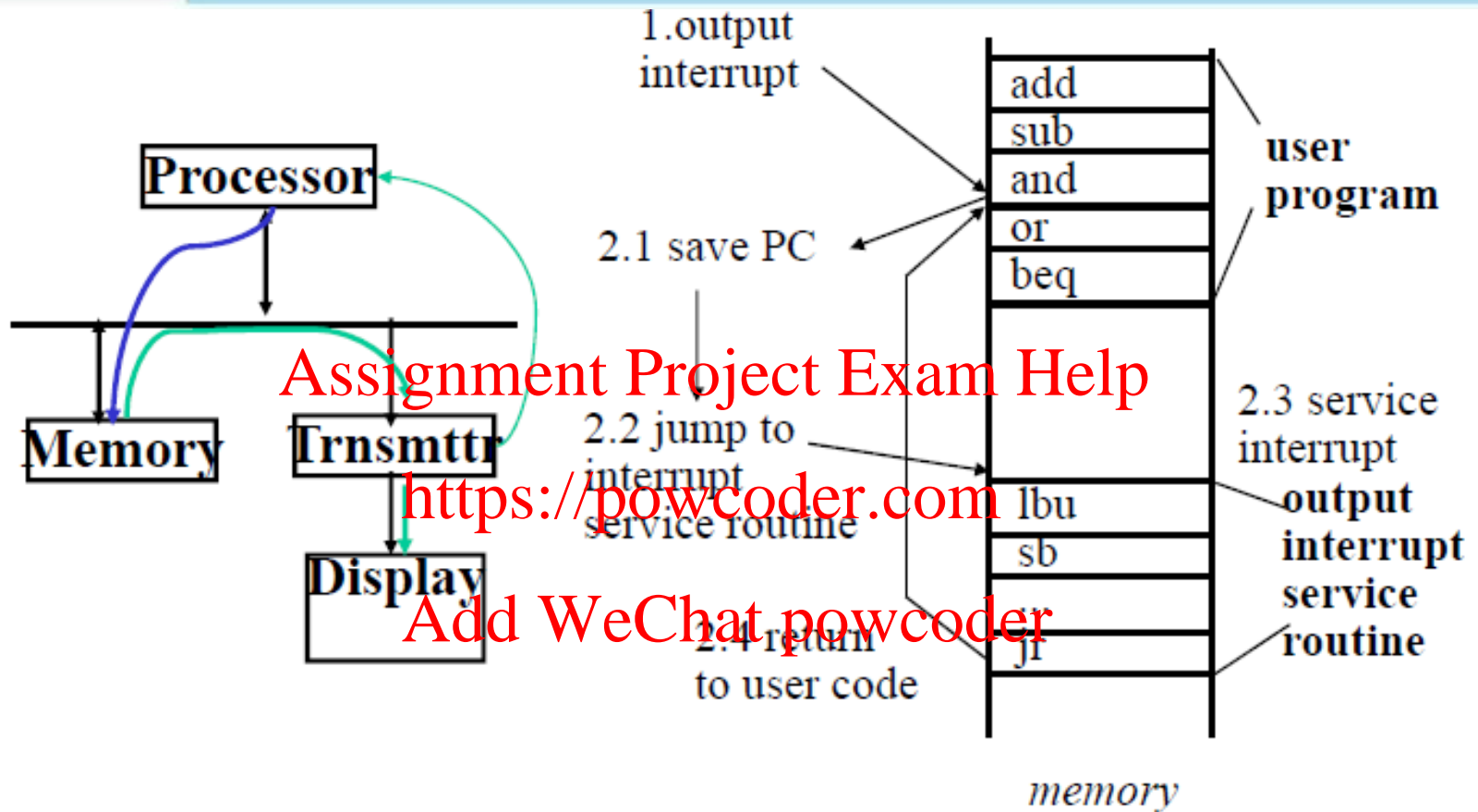
<https://powcoder.com>

Add WeChat powcoder

Interrupt Driven Output

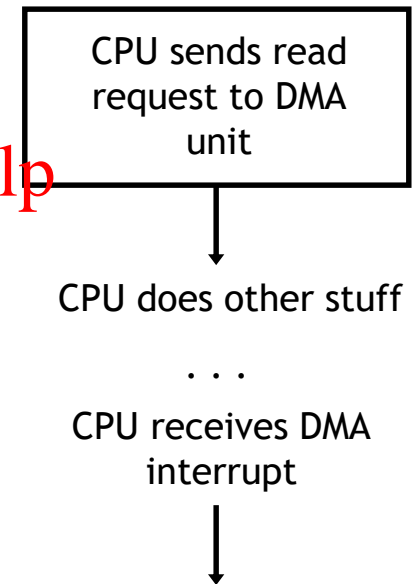


Interrupt Driven Output

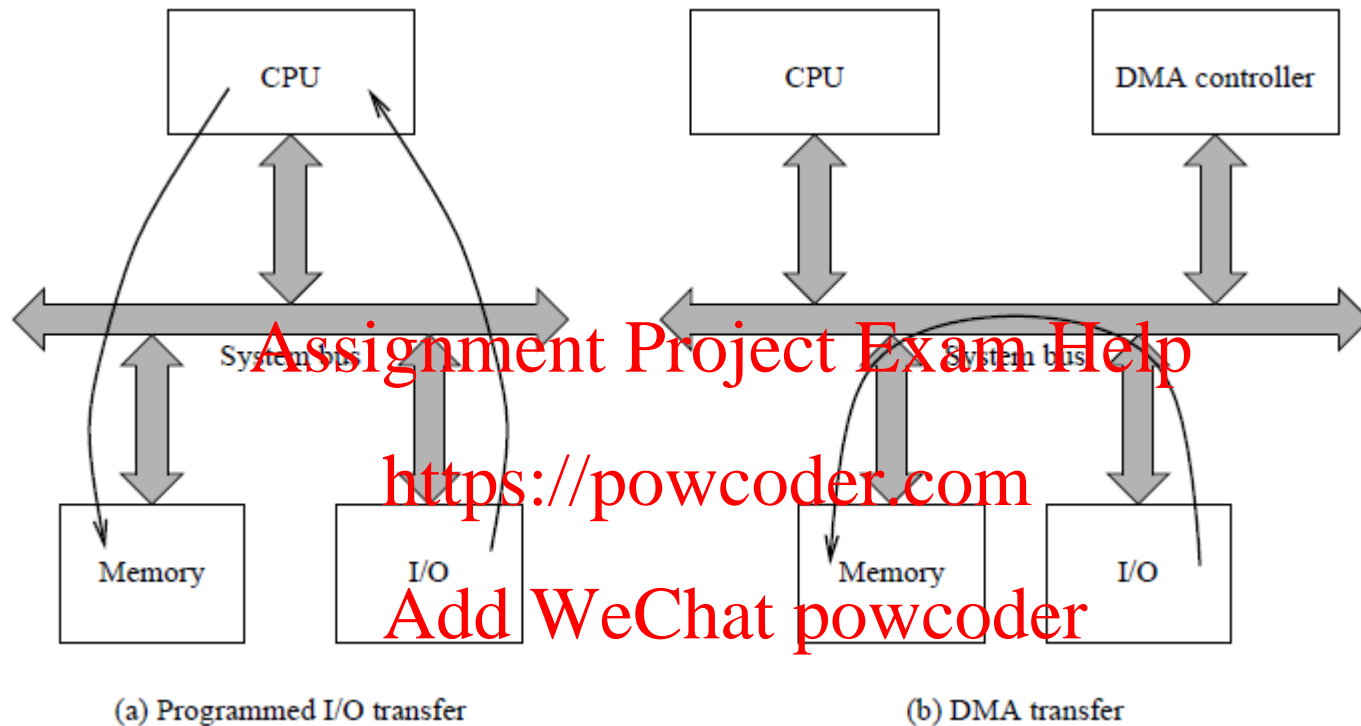


Direct memory access

- One final method of data transfer is to introduce a **direct memory access**, or **DMA**, controller
- The DMA controller is a simple processor which does most of the functions that the CPU would otherwise have to handle
 - The CPU asks the DMA controller to transfer data between a device and main memory. After that, the CPU can continue with other tasks
 - The DMA controller issues requests to the right I/O device, waits, and manages the transfers between the device and main memory
 - Once finished, the DMA controller interrupts the CPU
- DMA relieves the CPU of the low-level data transfer chore. But it needs hardware support as it is implemented by using a DMA controller.

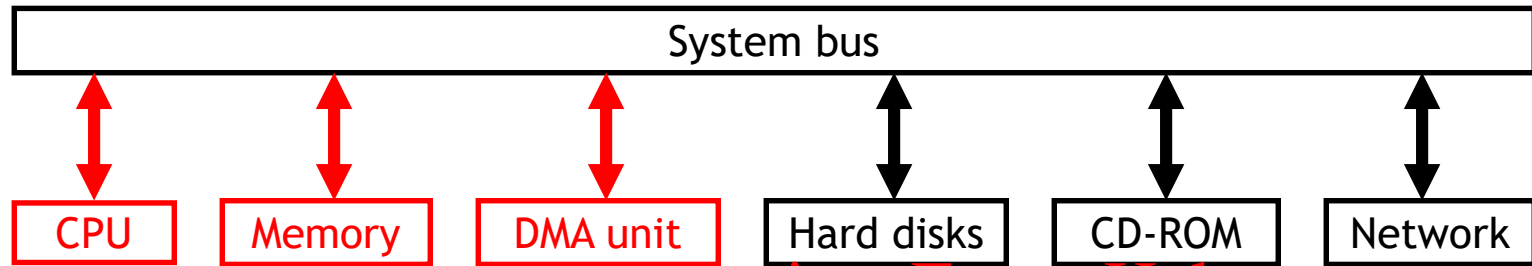


Programmed I/O vs DMA



- (a) In Programmed I/O, data are read by the processor and then written to the memory.
- (b) In DMA transfer, the DMA controller generates the control signals to transfer data directly between the I/O device and memory.

Direct memory access



Assignment Project Exam Help

<https://powcoder.com>

- Since both the processor and the DMA controller may need to access main memory, some form of arbitration is required

Add WeChat powcoder

Bus arbitration

- More than one bus master can request the bus
 - Need an arbitration mechanism to allocate the bus
- Bus arbitration can be done either
 - Statically
 - Dynamically
- Static arbitration
 - Done in a **predetermined way**
 - Easy to implement
 - Does not take needs into account
 - Poor utilization - Bus could be assigned even when not needed
- Dynamic bus arbitration
 - Bus allocated only **in response to a request**
 - Each master is equipped with
 - Bus request line
 - Bus grant line
 - A master uses the bus request line to let others know that it needs the bus
 - Before a master can use the bus, it must receive permission to use the bus via the bus grant line

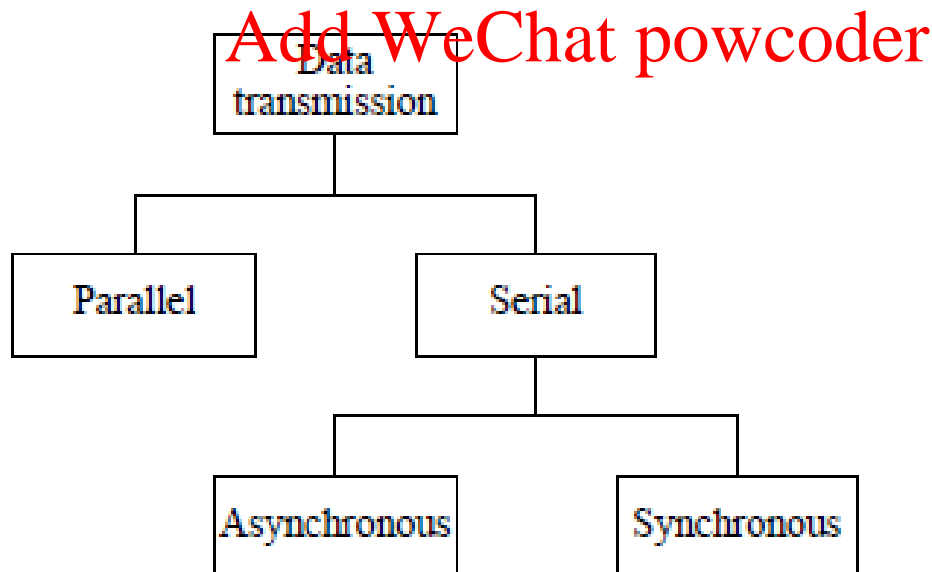
Assignment Project Exam Help

<https://powcoder.com>

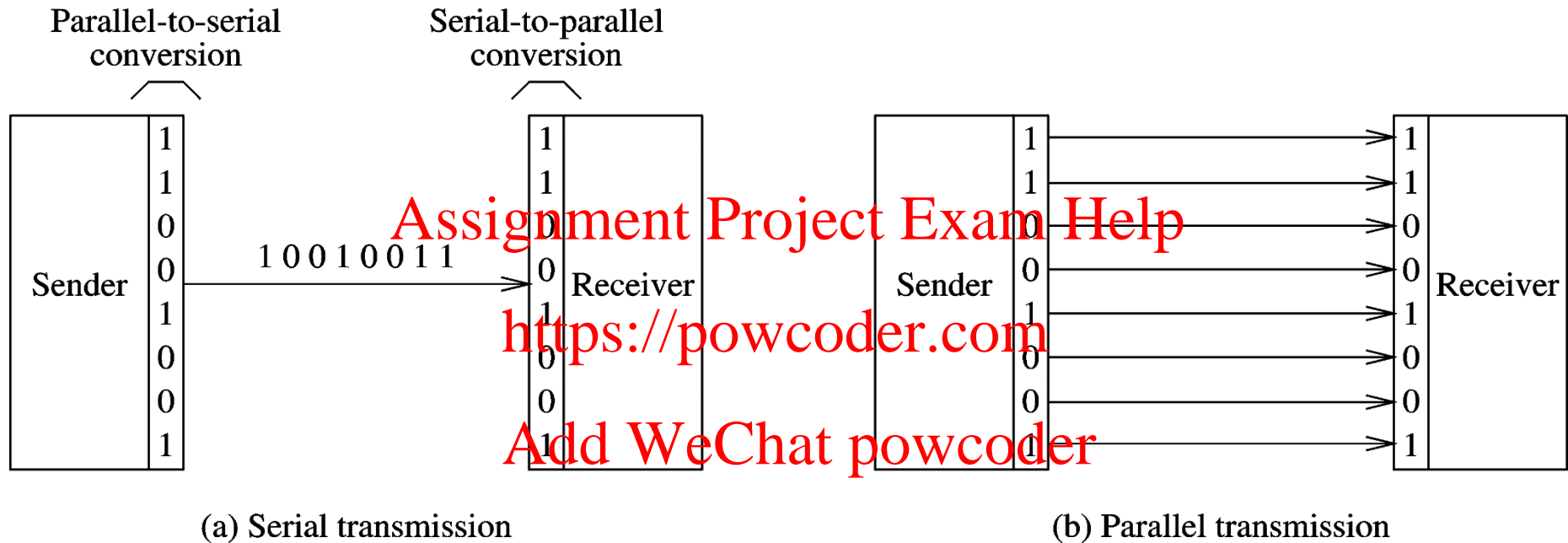
Add WeChat powcoder

Serial Communication

- Transfer of data may be parallel or serial.
- **Parallel data transfer:**
 - Each bit has its own path, and the entire message is transmitted at one time.
 - Faster, because multiple signal lines operate in parallel.
 - Used for short distances and when speed is important.
 - Data skew problem.
- **Serial data transfer:**
 - Each bit is sent in sequence, one at a time.
 - Slower, but less expensive since it requires only one conductor.
 - Become increasingly important: Ease of connecting smaller cables, and less signal skew problem as data rates increase.



Serial Communication



Serial Communication

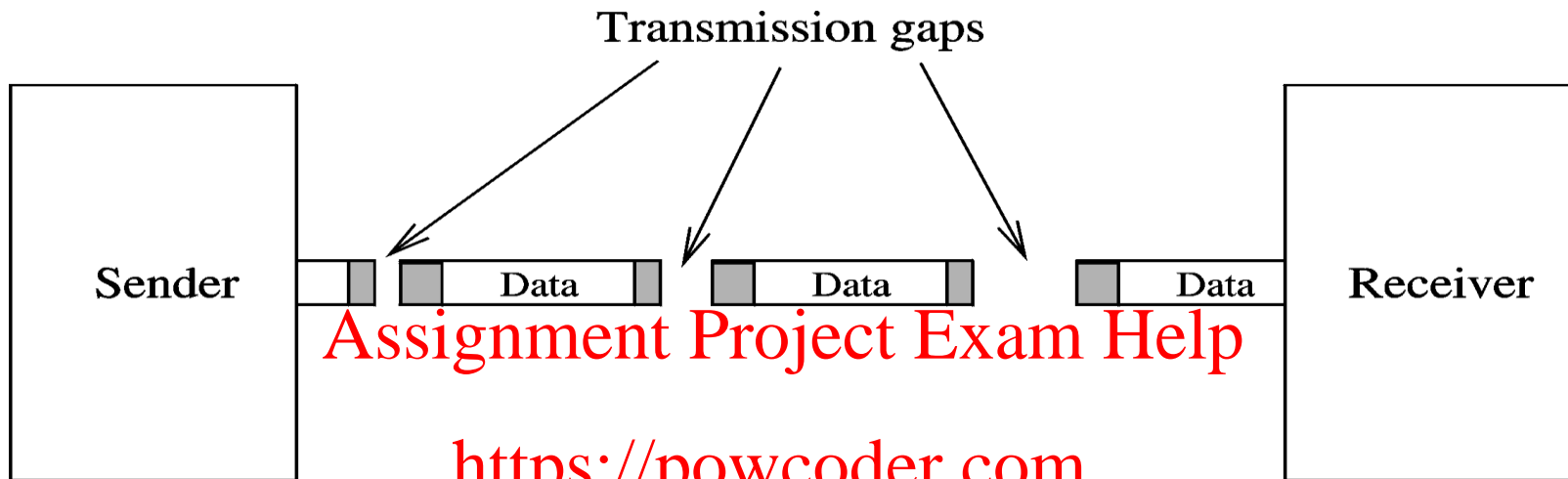
- Serial transmission can be asynchronous or synchronous.
- **Asynchronous transmission**
 - Each byte is encoded for transmission
 - Start and stop bits.
 - No need for sender and receiver synchronisation.
- **Synchronous transmission**
 - Sender and receiver must synchronise.
 - Done in hardware using phase locked loops (PLLs)
 - Block of data can be sent
 - More efficient
 - Less overhead than asynchronous transmission
 - Expensive
- Universal Serial Bus (USB) and Firewire use the synchronous transmission.

Assignment Project Exam Help

<https://powcoder.com>

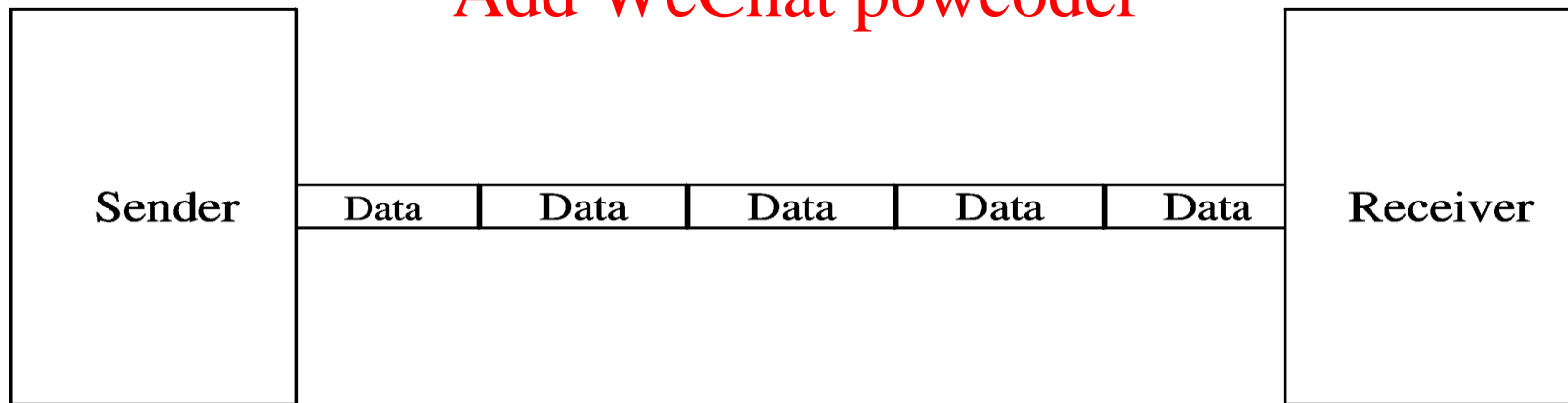
Add WeChat powcoder

Serial Communication



(a) Asynchronous transmission

Add WeChat powcoder



(b) Synchronous transmission

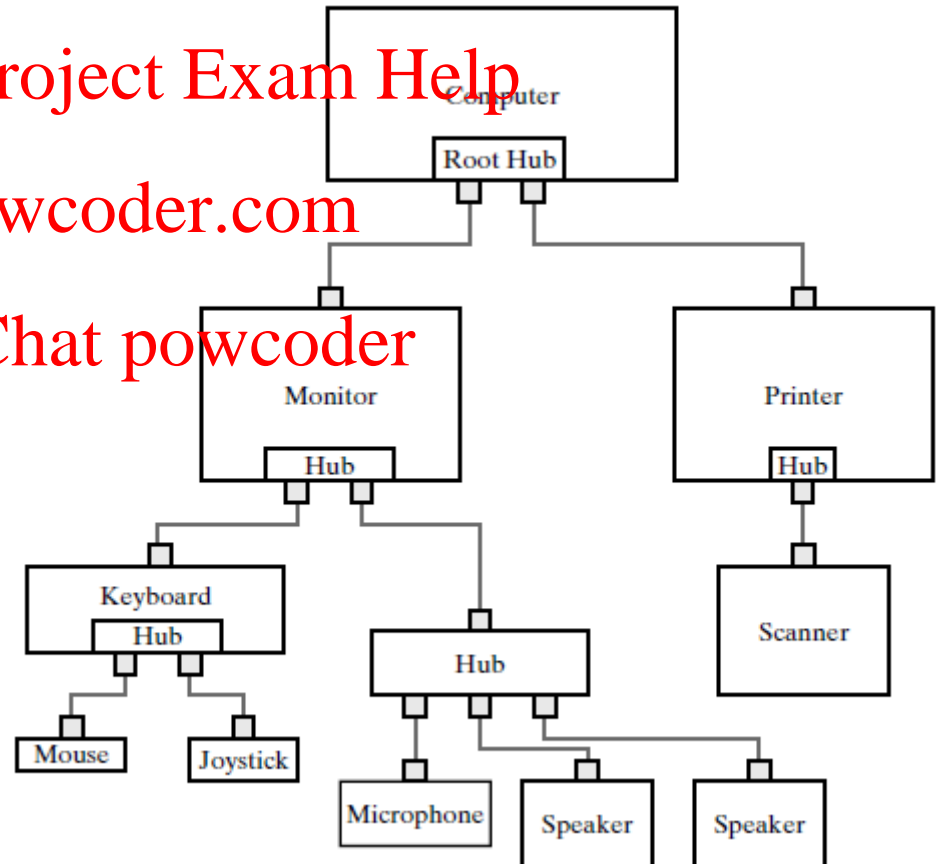
Universal Serial Bus (USB)

- Intended to help retire all legacy varieties of serial and parallel ports.
 - Single standardized interface socket
 - Plug-and-play + hot swapping
 - Providing power to low-consumption devices
 - Allowing many standard devices to be used without requiring manufacturer specific

□ The computer and attached devices can be classified as hubs, devices, or compound devices.

□ A **hub** provides attachment points for USB devices and other hubs. It contains a USB interface for control and status handling.

□ If a hub is combined with a device such as the keyboard in the figure, the keyboard is referred to as **compound devices**.



Interrupts and Exceptions

- **Interrupts** are external events that require the processor's attention
 - *Examples:* I/O device needs attention, timer interrupts to mark cycle
- The mechanism of interrupt is similar to **procedure call**, but interrupts can be initiated by both software and hardware.
 - Procedures are purely software-initiated.

Assignment Project Exam Help

- Interrupts are normal, not errors
 - All interrupts are recoverable: interrupted program will need to be resumed after the interrupt is handled.
- The MIPS processor uses the system call (`syscall`) instruction for software interrupts.
 - We used `syscall` to invoke the SPIM simulator I/O services.
- The interrupt service routine is commonly called *interrupt handler*.

Interrupts and Exceptions

- **Exceptions** are typically errors that are detected within the processor
 - *Examples:* illegal instruction opcode, arithmetic overflow, or attempts to divide by 0.
- There are two possible ways of resolving these errors:
 - If the error is **un-recoverable**, the operating system kills the program
 - Less serious problems can often be fixed by OS or the program itself
- The underlying mechanism of interrupts and exceptions is essentially the same.
- OS responsible to do the right thing, such as:
 - Save the current state and shut down the hardware devices
 - Find and load the correct data from the hard disk
 - Transfer data to/from the I/O device, or install drivers

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Interrupt Processing in MIPS

- On interrupts
 - Disables further interrupts
 - Record the cause of interrupts/exception – Cause register
 - Transfers control to exception handler located at a 0x80000180 (fixed address)
 - Handler saves the context
 - Program counter (PC)
 - Current operating mode (user or supervisor)
 - Status of interrupts (enables or disabled)
 - Stores return address
 - Uses a register: EPC (exception program counter) register

Exception	Cause
Hardware Interrupt	0x00000000
System Call	0x00000020
Breakpoint / Divide by 0	0x00000024
Undefined Instruction	0x00000028
Arithmetic Overflow	0x00000030

MIPS exception cause codes

Interrupt Processing in MIPS

- **Cause** and **EPC** registers are loaded into **k0** and **k1** registers

```
mfc0    $k0,$13    # copy Cause register (Register 13) into $k0
mfc0    $k1,$14    # copy EPC register (Register 14) into $k1
```

- Cause and EPC registers are not part of the MIPS register file.
 - Located in coprocessor0 (CP0)

Assignment Project Exam Help

- The processor returns to the address in **EPC** after handling the exception.
 - Return address = address of the interrupted instruction.
 - We need to add 4 to get to the instruction following the interrupted instruction

<https://powcoder.com>

Add WeChat powcoder

```
addiu    $k1,$k1,4
rfe
jr       $k1
```