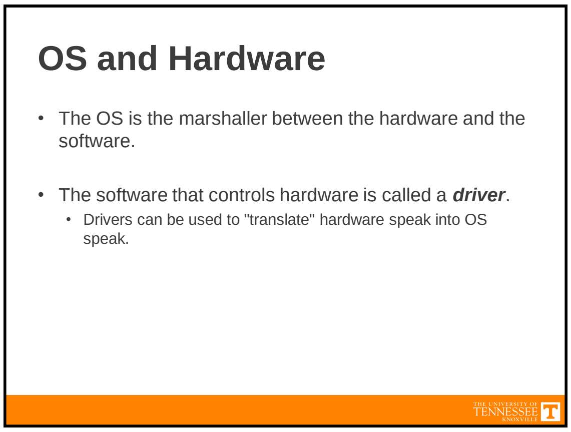
Drivers and Hardware

COSC-361
Stephen Marz

TICKLE COLLEGE OF ENGINEERING
MIN H. KAO DEPARTMENT OF ELECTRICAL ENGINEERING & COMPUTER SCIENCE

1

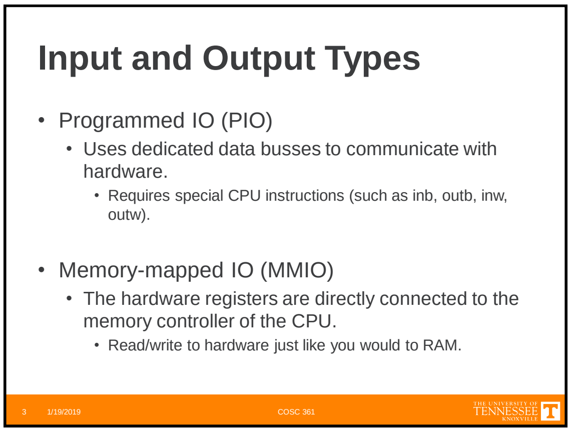


OS and Hardware

- The OS is the marshaller between the hardware and the software.
- The software that controls hardware is called a ***driver***.
  - Drivers can be used to "translate" hardware speak into OS speak.

THE UNIVERSITY OF TENNESSEE KNOXVILLE

2

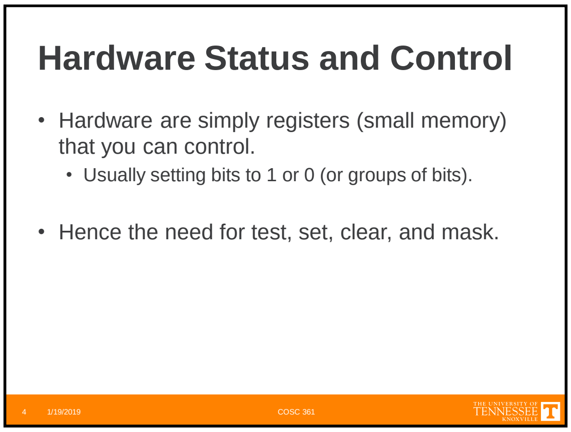


Input and Output Types

- Programmed IO (PIO)
  - Uses dedicated data busses to communicate with hardware.
    - Requires special CPU instructions (such as inb, outb, inw, outw).
- Memory-mapped IO (MMIO)
  - The hardware registers are directly connected to the memory controller of the CPU.
    - Read/write to hardware just like you would to RAM.

3 1/19/2019 COSC 361 THE UNIVERSITY OF TENNESSEE KNOXVILLE

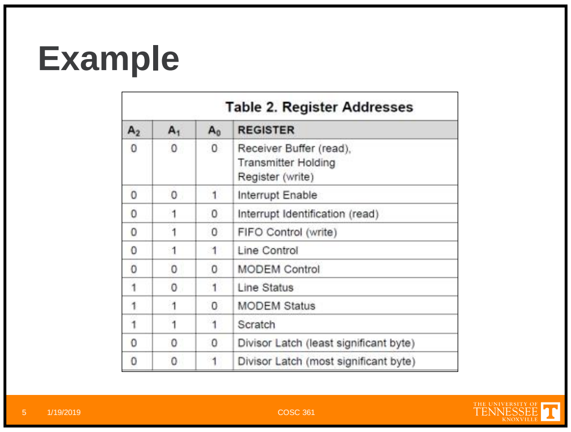
3



Hardware Status and Control

- Hardware are simply registers (small memory) that you can control.
  - Usually setting bits to 1 or 0 (or groups of bits).
- Hence the need for test, set, clear, and mask.

4

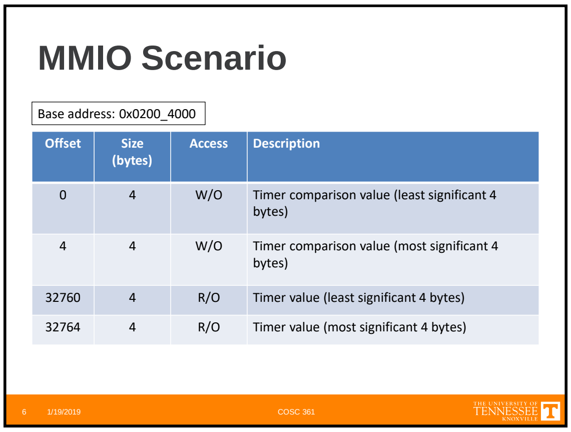


Example

Table 2. Register Addresses

A_2	A_1	A_0	REGISTER
0	0	0	Receiver Buffer (read), Transmitter Holding Register (write)
0	0	1	Interrupt Enable
0	1	0	Interrupt Identification (read)
0	1	0	FIFO Control (write)
0	1	1	Line Control
0	0	0	MODEM Control
1	0	1	Line Status
1	1	0	MODEM Status
1	1	1	Scratch
0	0	0	Divisor Latch (least significant byte)
0	0	1	Divisor Latch (most significant byte)

5

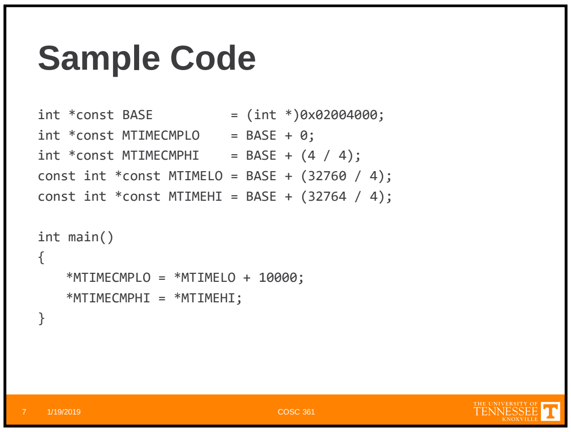


MMIO Scenario

Base address: 0x0200_4000

Offset	Size (bytes)	Access	Description
0	4	W/O	Timer comparison value (least significant 4 bytes)
4	4	W/O	Timer comparison value (most significant 4 bytes)
32760	4	R/O	Timer value (least significant 4 bytes)
32764	4	R/O	Timer value (most significant 4 bytes)

6

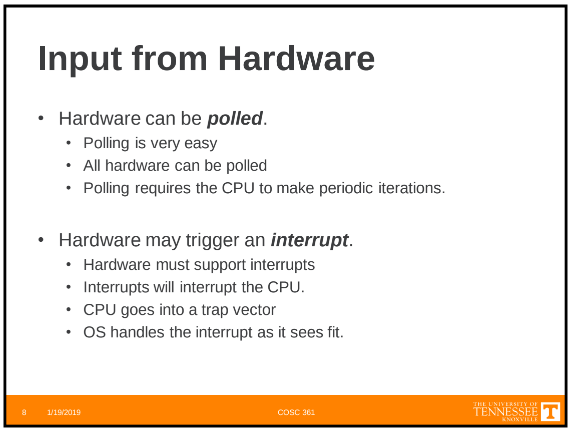


Sample Code

```
int *const BASE          = (int *)0x02004000;
int *const MTIMECMPLO    = BASE + 0;
int *const MTIMECMPHI    = BASE + (4 / 4);
const int *const MTIMELO = BASE + (32760 / 4);
const int *const MTIMEHI = BASE + (32764 / 4);

int main()
{
    *MTIMECMPLO = *MTIMELO + 10000;
    *MTIMECMPHI = *MTIMEHI;
}
```

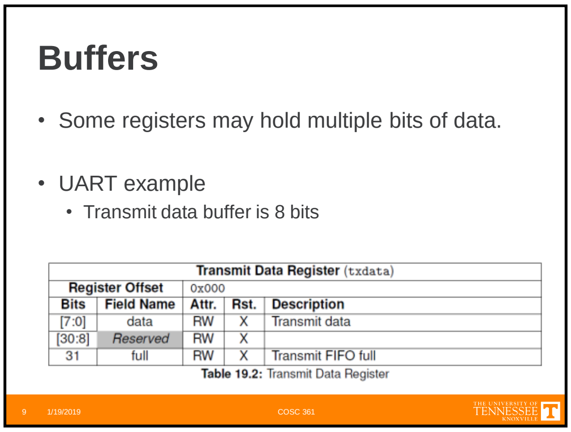
7



Input from Hardware

- Hardware can be ***polled***.
  - Polling is very easy
  - All hardware can be polled
  - Polling requires the CPU to make periodic iterations.
- Hardware may trigger an ***interrupt***.
  - Hardware must support interrupts
  - Interrupts will interrupt the CPU.
  - CPU goes into a trap vector
  - OS handles the interrupt as it sees fit.

8



Buffers

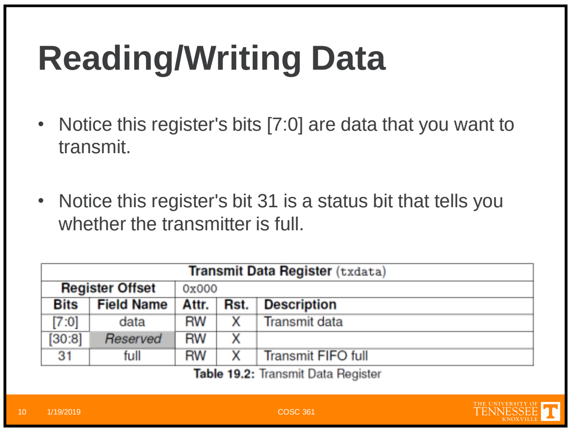
- Some registers may hold multiple bits of data.
- UART example
  - Transmit data buffer is 8 bits

Transmit Data Register (`txdata`)

Register Offset		`0x000`		
Bits	**Field Name**	**Attr.**	**Rst.**	**Description**
[7:0]	data	RW	X	Transmit data
[30:8]	*Reserved*	RW	X	
31	full	RW	X	Transmit FIFO full

Table 19.2: Transmit Data Register

9



Reading/Writing Data

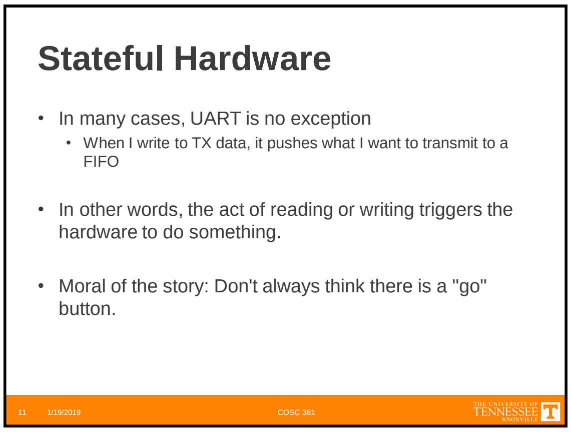
- Notice this register's bits [7:0] are data that you want to transmit.
- Notice this register's bit 31 is a status bit that tells you whether the transmitter is full.

Transmit Data Register (`txdata`)				
Register Offset	0x000			
Bits	**Field Name**	**Attr.**	**Rst.**	**Description**
[7:0]	data	RW	X	Transmit data
[30:8]	*Reserved*	RW	X	
31	full	RW	X	Transmit FIFO full

Table 19.2: Transmit Data Register

10 1/19/2019 COSC 361

10

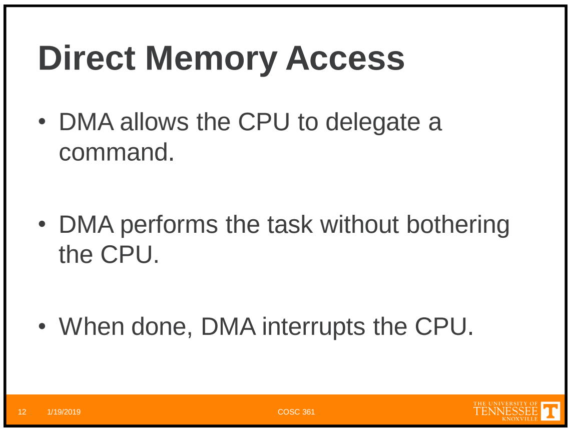


Stateful Hardware

- In many cases, UART is no exception
  - When I write to TX data, it pushes what I want to transmit to a FIFO
- In other words, the act of reading or writing triggers the hardware to do something.
- Moral of the story: Don't always think there is a "go" button.

11 1/19/2019 COSC 361

11



Direct Memory Access

- DMA allows the CPU to delegate a command.
- DMA performs the task without bothering the CPU.
- When done, DMA interrupts the CPU.

12 1/19/2019 COSC 361

12

Drivers & Hardware

COSC-361
Stephen Marz

THE UNIVERSITY OF TENNESSEE KNOXVILLE

13