

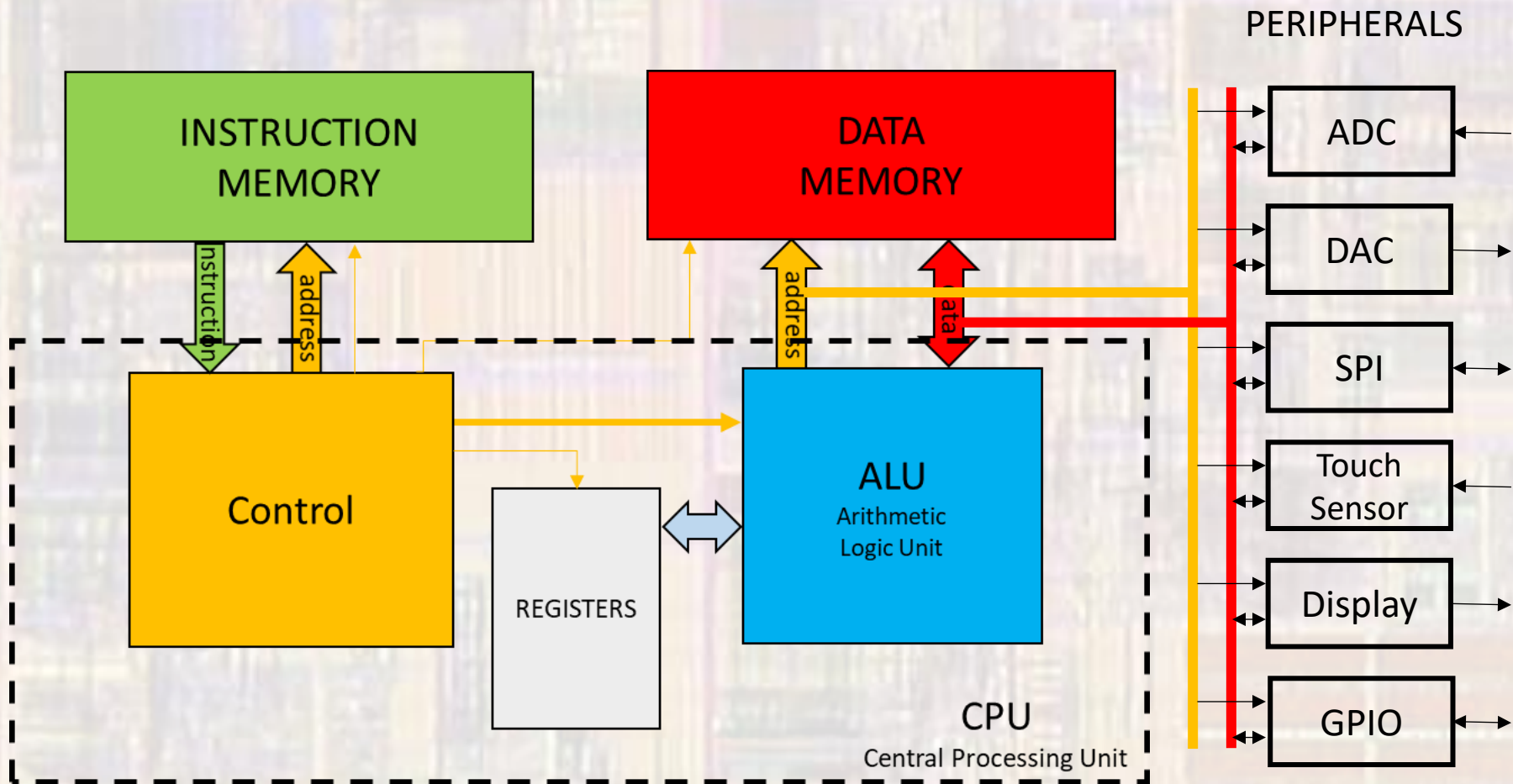
# Embedded Systems Examples

Last updated 7/25/24

These slides provide some examples of using embedded systems

# Embedded Systems Examples

- Embedded System Components
  - Peripherals are system dependent



# Embedded Systems Examples

- GPIO peripheral - setup
  - GPIO is used to control groups of external pins
  - GPIO registers
    - Data In register (8 bits)
    - Data Out register (8 bits)
    - Direction register (8 bits) – input(0), output(1)
  - Create a structure to access these 3 registers

```
typedef struct {  
    uint8_t data_in;  
    uint8_t data_out;  
    uint8_t dir;  
} GPIO ;
```

- GPIO\_A block base address = 0xF020
  - Create a pointer to the GPIO\_A block structure

```
GPIO * GPIOA = 0xF020
```

# Embedded Systems Examples

- Set all the pins in GPIO\_A to outputs and 1s

```
GPIOA->dir = 0xFF;           // set direction to out
GPIOA->data_out = 0xFF;      // set all outputs to 1
```

- Set just pins 5 and 1 to outputs without changing the direction of any other pins

```
GPIOA->dir |= 0x22;          // set pins 5 and 1 to out
```

- Set pin 6 to an input without changing the direction of any other pins

```
GPIOA->dir &= ~0x40;         // set pins 6 to input
```



# Embedded Systems Examples

- GPIOA setup as - in in in in out out out out
- Set pins 2,3 to 1s, and 0,1 to 0s

```
GPIOA->data_out |= 0x0C; // pins 3, 2 to 1  
GPIOA->data_out &= ~0x03; // pins 1, 0 to 0
```

- Read the value on pin 5

```
GPIOA->data_in & 0x20; // get only pin 5 value
```

- Set pin 0 to match the value on pin 7

```
if(GPIOA->data_in & 0x80)  
    GPIOA->data_out |= 0x01; // set pin 0  
else  
    GPIOA->data_out &= ~0x01; // clear pin 0
```

# Embedded Systems Examples

- ADC peripheral - setup
  - ADC converts the analog voltage on a pin to a digital value
  - ADC registers
    - Data register (16 bits)
    - Control register (8 bits)
      - start, stop, done, 12b/8b mode, input pin(3 bits – pins a-h), reserved
  - Create a structure to access these registers

```
typedef struct {  
    uint16_t data;  
    uint8_t ctrl;  
} ADC ;
```

- ADC\_1 block base address = 0xF0C0
  - Create a pointer to the ADC\_1 block structure

```
ADC * ADC1 = 0xF0C0
```

# Embedded Systems Examples

- Setup the ADC to convert the voltage on pin f in 12bit mode

```
////////////////////
// ADC setup for pin F
////////////////////
// adc ctrl : start, stop, done, mode, input pin (3b), N/A
// bits :      7      6      5      4      3-1      0
// mode: 1 – 12b, 0 – 8b
// input pin: 7-0 :: h-a
//
// stop any running conversion – set bit 6 in ctrl
ADC1->ctrl |= 0x40;
// select pin f – bits 3-1 in ctrl – pin f = 101
ADC1->ctrl &= ~0x0F;          // clear pin selection bits
ADC1->ctrl |= 0x0A;          // select pin f – 101x
// set mode to 12bit – set bit 4 in ctrl
ADC1->ctrl |= 0x10;
// release stop bit – clear bit 6 in ctrl
ADC1->ctrl &= ~0x40;
```

# Embedded Systems Examples

- Run a conversion

```
uint16_t result;

// run conversion
// adc ctrl : start, stop, done, mode, input pin (3b), N/A
// bits :      7   6   5   4   3-1   0
//
// start ADC
ADC1->ctrl |= 0x80;
// wait for done – bit 5 = 1
while(ADC1->ctrl & 0x20 == 0){
    // do nothing, not done
}
// read result
result = ADC1->data;
```