EE 357 Unit 10c

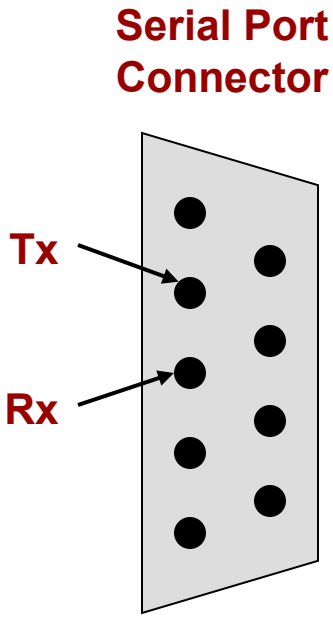
Serial I/O

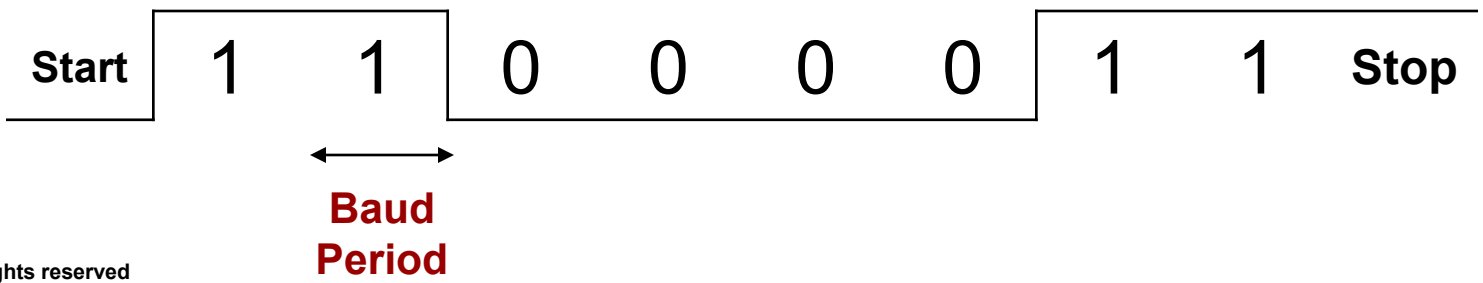
A-to-D Converter

DMA

RS-232 Serial Operation

- RS-232 Serial Ports communicate data 1-bit at a time at a certain baud rate (bits per second)
- Control bits (start/stop/parity) are usually included in data stream
- Full-duplex transmission
  - Transmit and Receive at the same time
  - 2 I/O pins: Tx and Rx
- Half-duplex: Only 1 transmitter at a time
- Example: transmission of 11000011

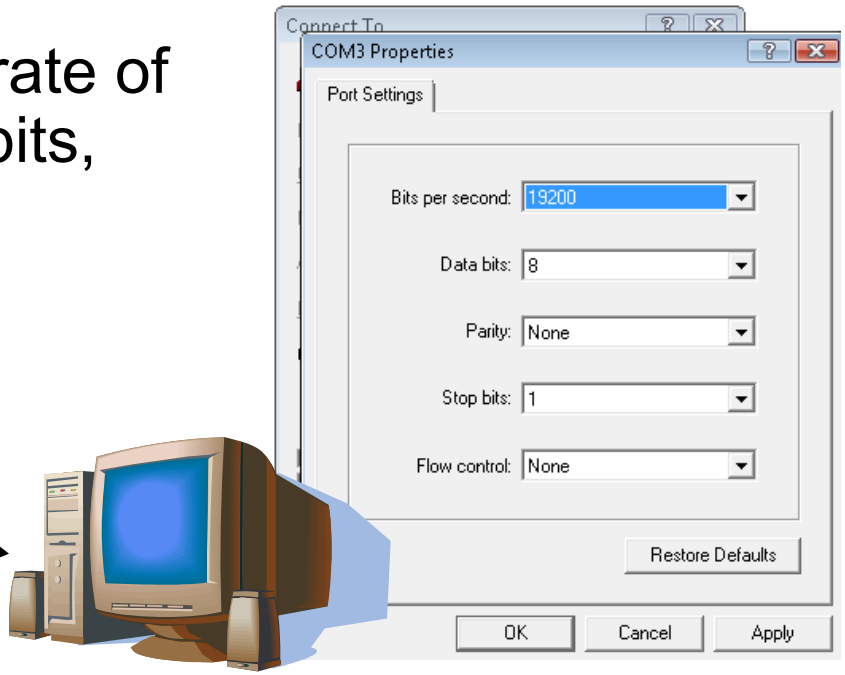




Establishing Serial Connections

- Both devices must be configured for the same settings to be able to communicate data correctly
- In our case, we will be communicating with a PC running a serial terminal program that allows us to configure our settings
- We will normally use a baud rate of 19200, 8 data bits, no parity bits, 1 stop bit, and no flow control

RS-232 Serial



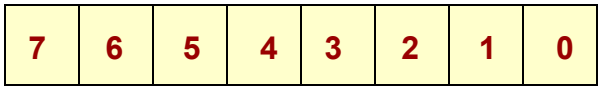
Serial settings controlled by configuration registers

Error Checking & Flow Control

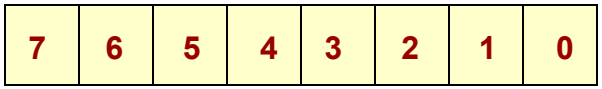
- Error-checking ensures that the data the receiver thinks it got is what the transmitter thinks it sent
  - A bit may get flipped or synchronization may be lost and bits are misinterpreted
  - Adding a parity bit allows us to detect an error (but not correct it) and ask the transmitter to re-send the data
  - Parity idea: Always have an even number of 1's in the transmission. The parity bit is whatever value is needed for an even # of 1's
- Flow control ensures that the receiver can keep up with the transmission rate of the transmitter
  - A receiver may have to execute other critical code and won't be able to drain the received data
  - Hardware method
    - RTS – Request to Send
    - CTS – Clear to Send

Serial Control

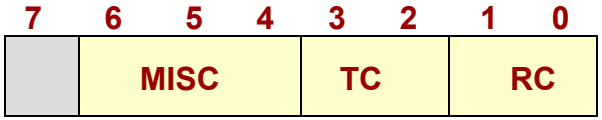
- Data registers
  - UTB – Transmit Buffer
  - URB – Receive Buffer
- Control Registers
  - UCR – UART Command Register
    - TC/RC: Enable (01) / Disable (10) Tx or Rx
    - MISC: Reset TX, RX, Mode Register Pointer
  - UMR1/2 – UART Mode Reg.
    - 2 reg.'s (1/2) share the same address
    - 1st write = UMR1, 2nd write = UMR2
    - To get back to UMR1, must issue command via UCR
    - PM & PT = Parity bit configuration
    - B/C = Bits per character = Data bits
    - SB = Number of stop bits
    - CM = Echo/Loopback (Should be disabled)



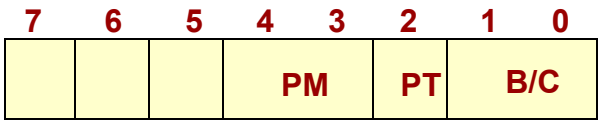
UTB = Data to be transmitted



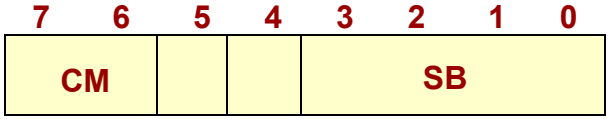
URB = Data received



UCR = Command Reg.



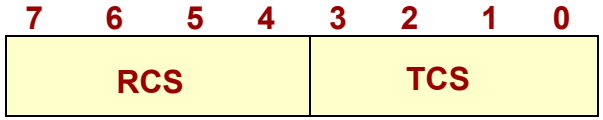
UMR1 = Mode Reg. 1



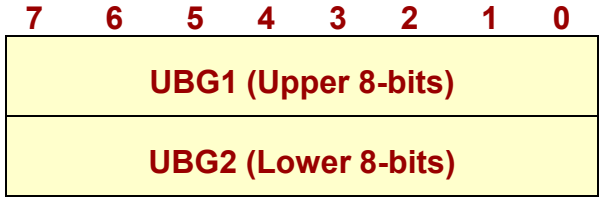
UMR2 = Mode Reg. 2

Serial Control (Cont.)

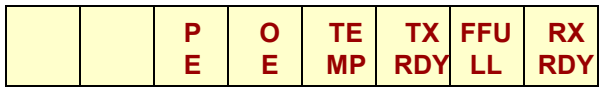
- Control Registers
  - UCSR – Select Rx and Tx clock sources for baud clock. Options are:
    - SYS_CLK (this is what we want)
    - External clock generators
  - UBG1/2 – Baud rate registers
    - (2) 8-bit reg's form a 16-bit UBG value
    - Desired Baud Rate = (F_{SYSCLK}/2) / (32 * UBG)
  - USR – Status Register
    - PE (Parity Error) / OE (Overrun Error)
    - TXRDY = Transmitter RDY (OK to write next char to UTB)
    - RXRDY = Receiver RDY (Data received and ready to be read from URB)
  - UIMR – Interrupt Mask/Status Register
    - Can generate interrupts when TXRDY/RXRDY (we will not use this)



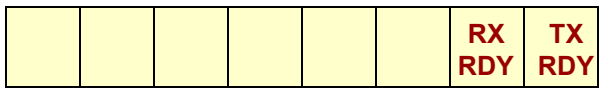
UCSR = Clock Source Reg.



UBG1/2 = Baud Generator



USR = Status Register



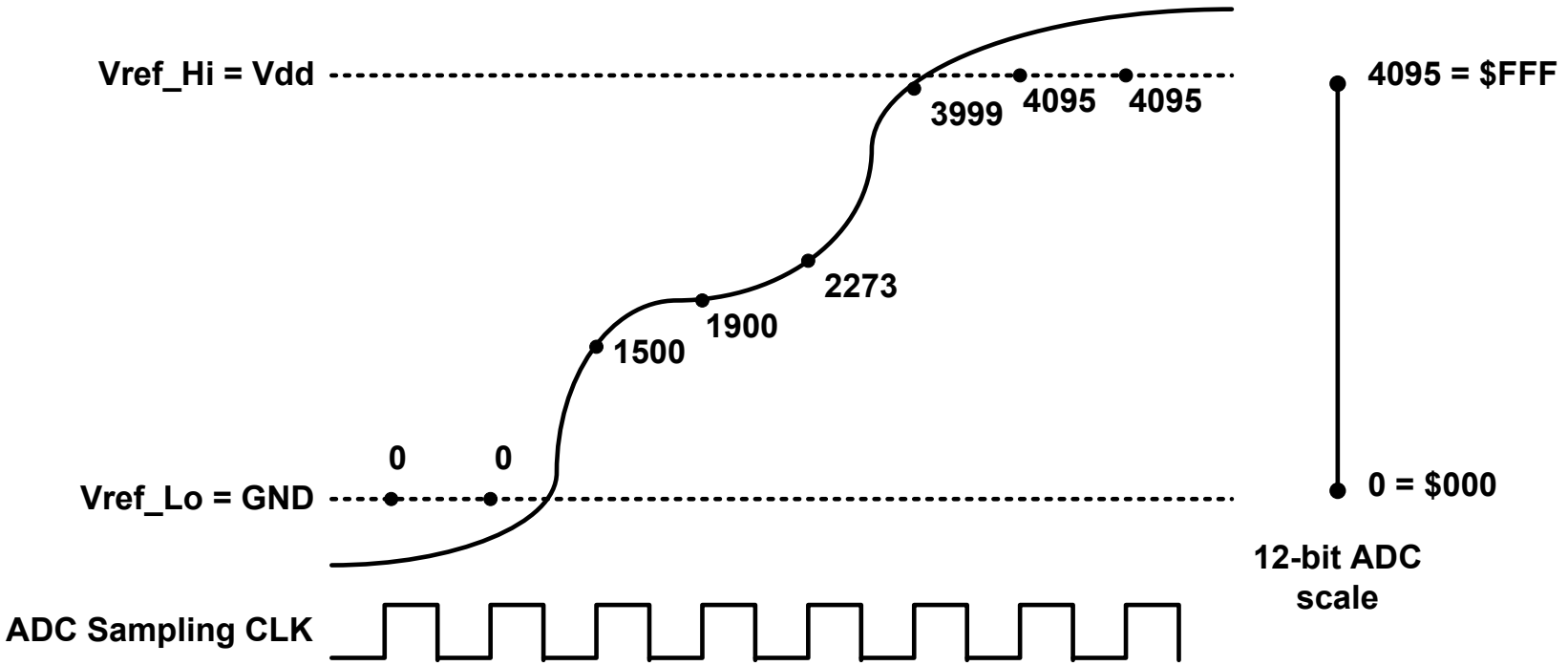
UIMR = Interrupt Mask/Status Register

UART Usage

- Initialization
  - Set PORTUA pins for UART functionality
  - Reset Tx, Rx, Mode Register via UCR
  - Configure serial operation via UMR1/2
  - Set Rx/Tx clock sources
  - Mask/Disable Interrupts
  - Set baud rate via UBG1/2
  - Enable Rx, Tx via UCR
- Operation (for transmitting and receiving)
  - Poll TXRDY (in USR) then write data to UTB
  - Poll RXRDY (in USR) then read data from URB

Analog to Digital Converters (ADC)

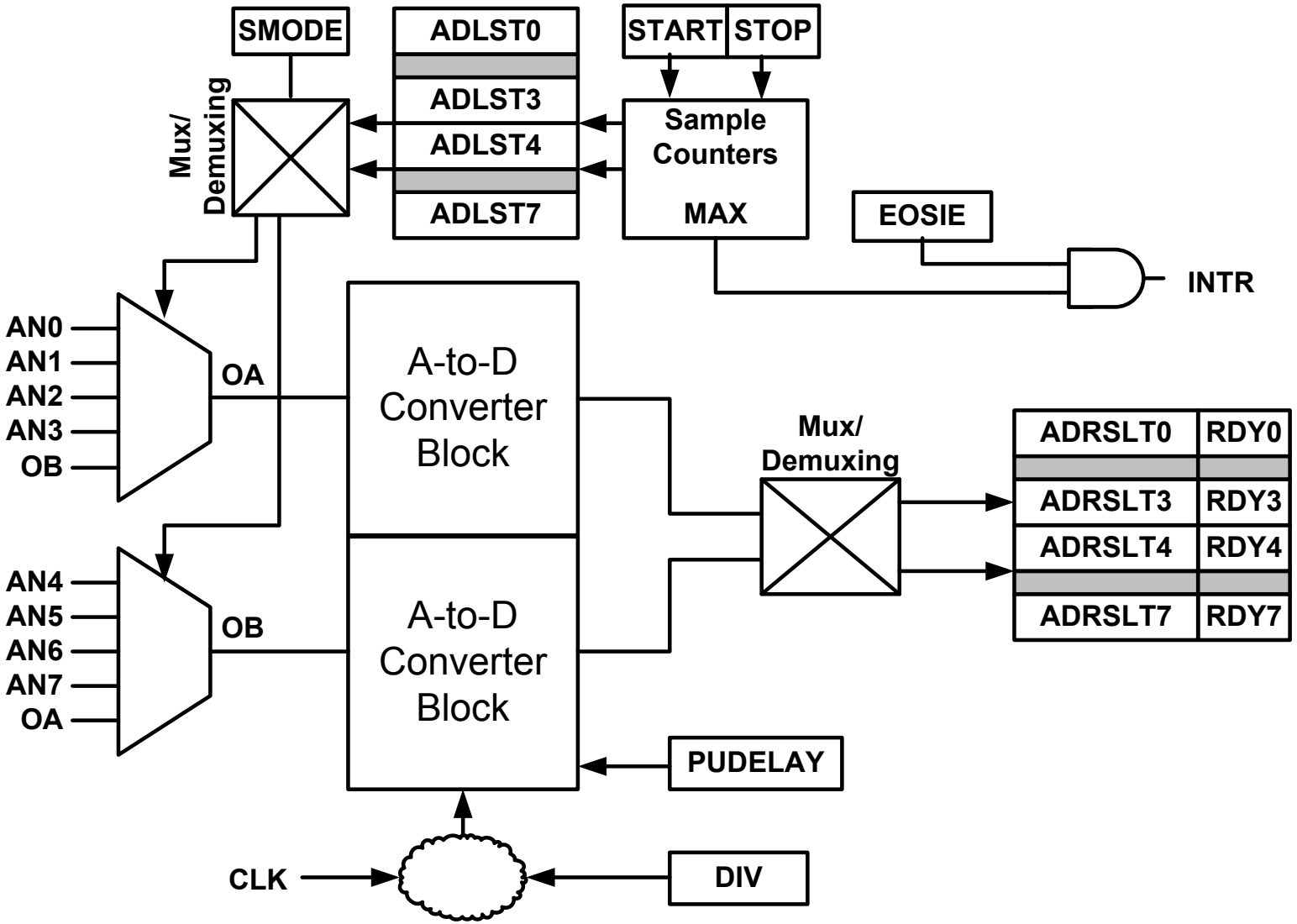
- Converts a voltage between V_{ref_Lo} and V_{ref_Hi} to a digital number
  - Saturates at max and min value



MCF52259 A-to-D Operation

- 8 Input channels (analog inputs) multiplexed to 2 ADC blocks
  - Sequential = Use only 1 ADC block
  - Parallel = Use both ADC blocks simultaneously
- Sampled at SW definable sampling rate / clock
- Single-Ended (1 signal measured on an absolute voltage scale) or Differential (voltage difference between 2 signals is measured)
- Takes samples in groups called a “scan”
  - Up to 8 samples in a scan and can be any mix of channels (all 8 samples from 1 channel, 1 from each of the 8 sources, anything in between)
- Once, triggered, or looping scans
  - Take one scan of 8 based when a user sets a “start” signal, can take a scan on a trigger, or can automatically restart a scan when previous one ends

A-to-D Converter Logic

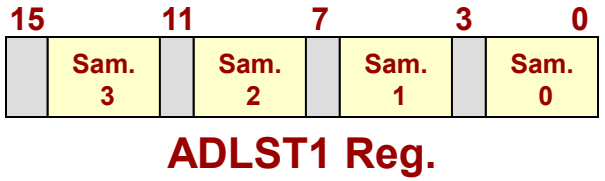


ADC Settings

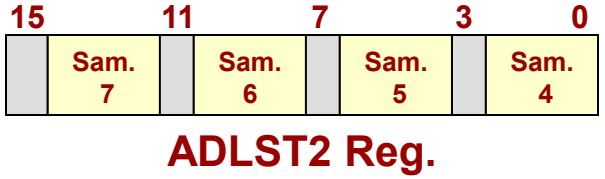
- Control 1 Register
  - START/STOP = Start a scan by writing a '1' to start (but STOP must be '0' or else the start command will be ignored. Stop bit is set to '1' automatically at the end of a scan.
  - SMODE = Sets Sequential/Parallel scanning and Once, Triggered, Looping scans (we will usually use Once/Sequential)
  - EOSIE (End-of-Scan Interrupt Enable) = Can be used to know when to read results via an interrupt handler
- Control 2 Register
  - DIV = Clock divider factor for sampling clock generation
- Power Register
  - PUDELAY (Power-up Delay) = number of clocks to wait for ADC calibration when powered up (use default of 4)

ADC Settings

- ADLST0-7: 3-bit fields for each of the 8 samples indicating which channels should be sampled
  - 8 samples from channel 1: ADLST0-7 all equal to 1
  - 1 sample from each channel in order: ADLST0-7 set to 0-7 respectively
  - 4 samples from channel 0 then 4 from channel 5: ADLST0-3 = 0, ADLST4-7=5
- ADRSLT0-7: Results from sample 0-7 of the scan
  - Need to be shifted right by 3-bits
  - SEXT (Sign Extend) = '1' if we want a signed number but we usually just want an unsigned number ('0')
- ADSTAT Register
  - RDY0-7 = Used for polling with each bit indicating the result in ADRSLTx is complete and ready for reading
  - EOSI0 = Interrupt flag for End-of-Scan (Acknowledge/clear by writing a '1')



ADLST1 Reg.



ADLST2 Reg.

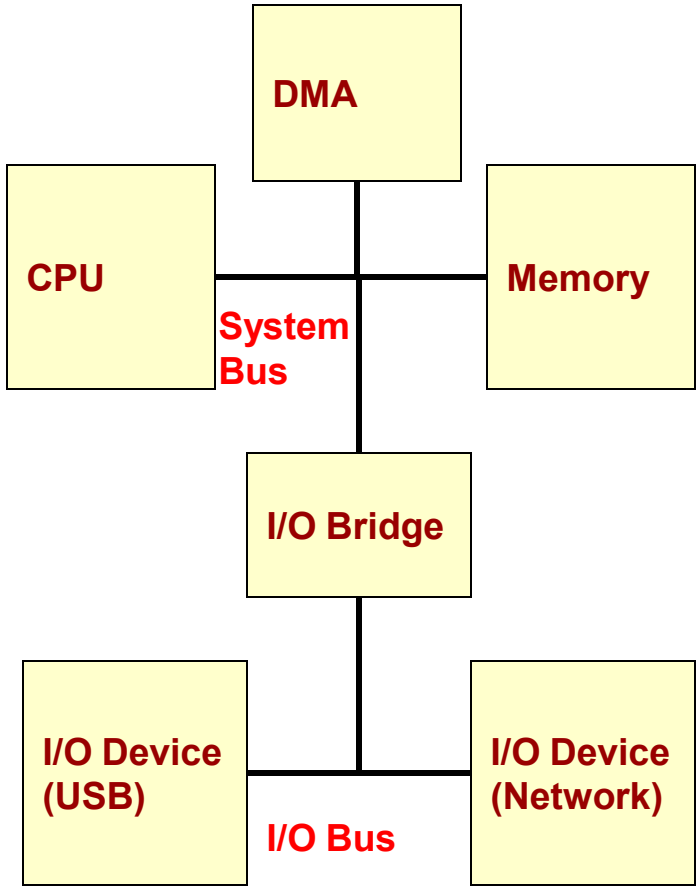
ADRSLTx

ADC Usage

- Initialization
  - Set PORTAN I/O pins for ADC function
  - Initialize SMODE in CTRL1
  - Initialize clock divider in CTRL2
  - Initialize Power-Up Delay
  - Set ADLST with desired channels
- Operation
  - Clear STOP bit
  - Set START bit
  - Poll appropriate RDY bits
  - Read ADRSLTx registers and shift right by 3

Direct Memory Access (DMA)

- Large buffers of data often need to be copied between:
  - Memory and I/O (video data, network traffic, etc.)
  - Memory and Memory (OS space to user app. space)
- DMA devices are small hardware devices that copy data from a source to destination freeing the processor to do "real" work

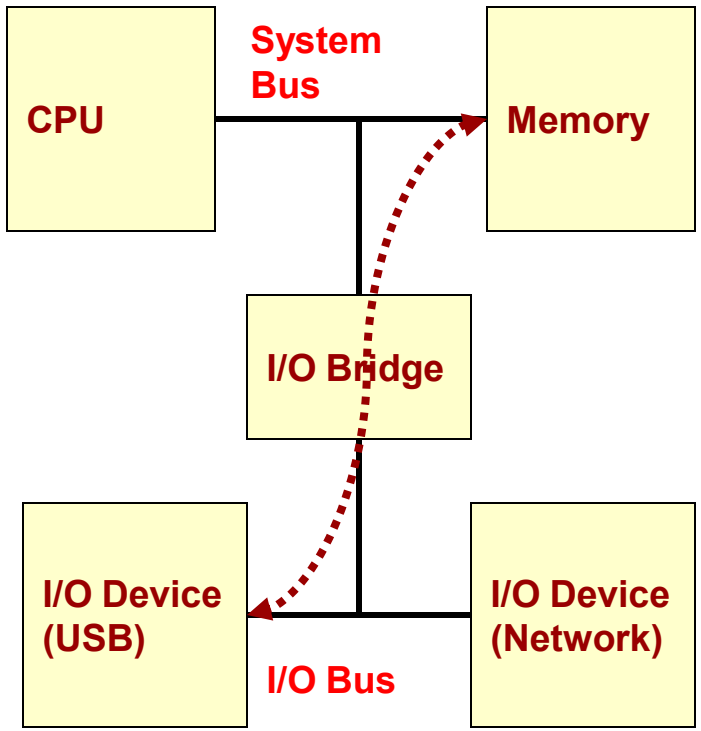


Data Transfer w/o DMA

- Without DMA, processor would have to move data using a loop
- Move 16K longwords pointed to by A0 to A1

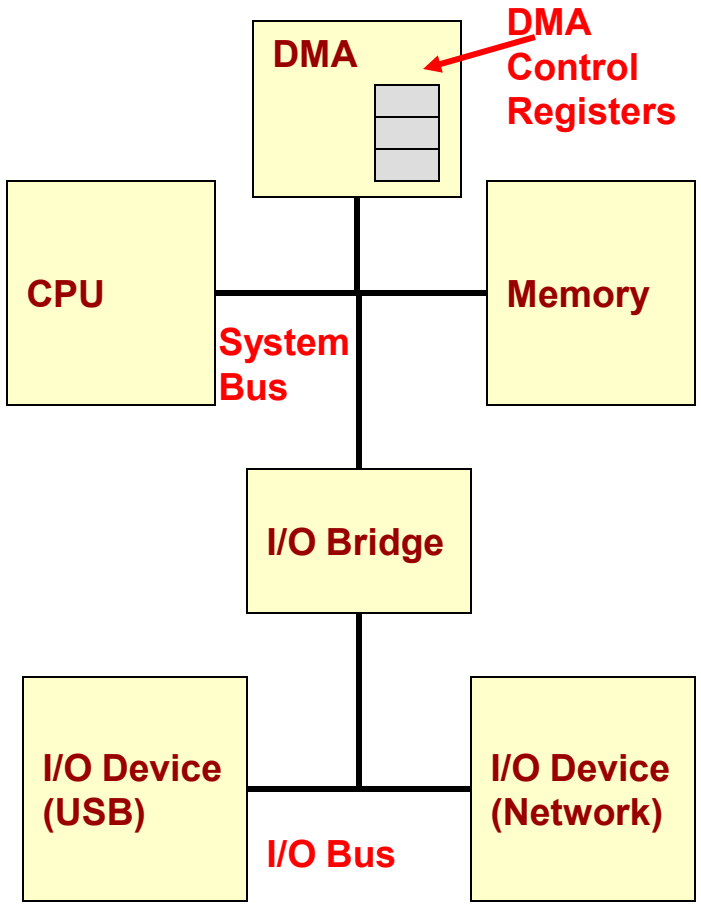
```
        move.l  #16384,d0
AGAIN:  move.w  (a0)+,(a1)+
        subq.l  #1,d0
        bne     AGAIN
```

- Processor wastes valuable execution time moving data



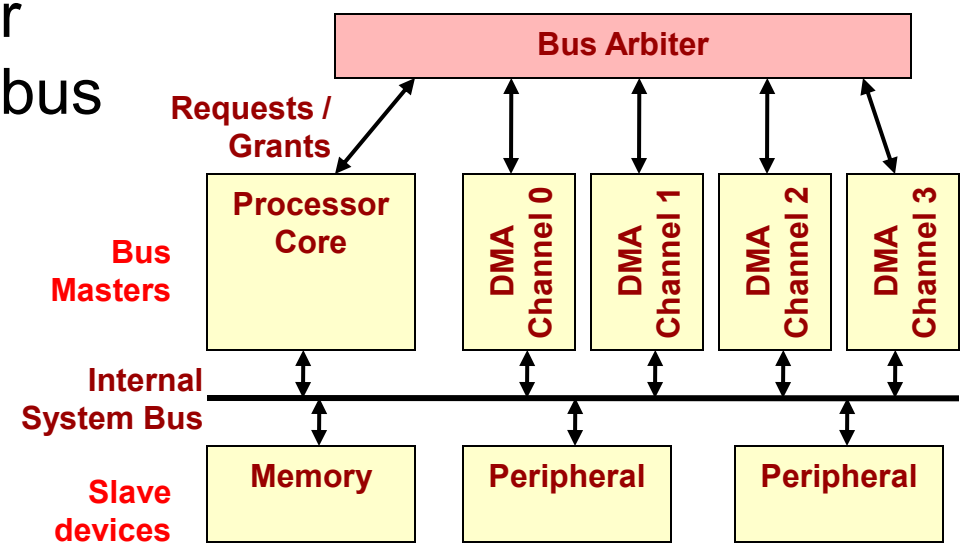
Data Transfer w/ DMA

- Processor sets values in DMA control registers
  - Source Start Address
  - Dest. Start Address
  - Byte Count
  - Control & Status (Start, Stop, Interrupt on Completion, etc.)
- DMA becomes "bus-master" (controls system bus to generate reads and writes) while processor is free to execute other code
  - Small problem: Bus will be busy
  - Hopefully, data & code needed by the CPU will reside in the processor's cache



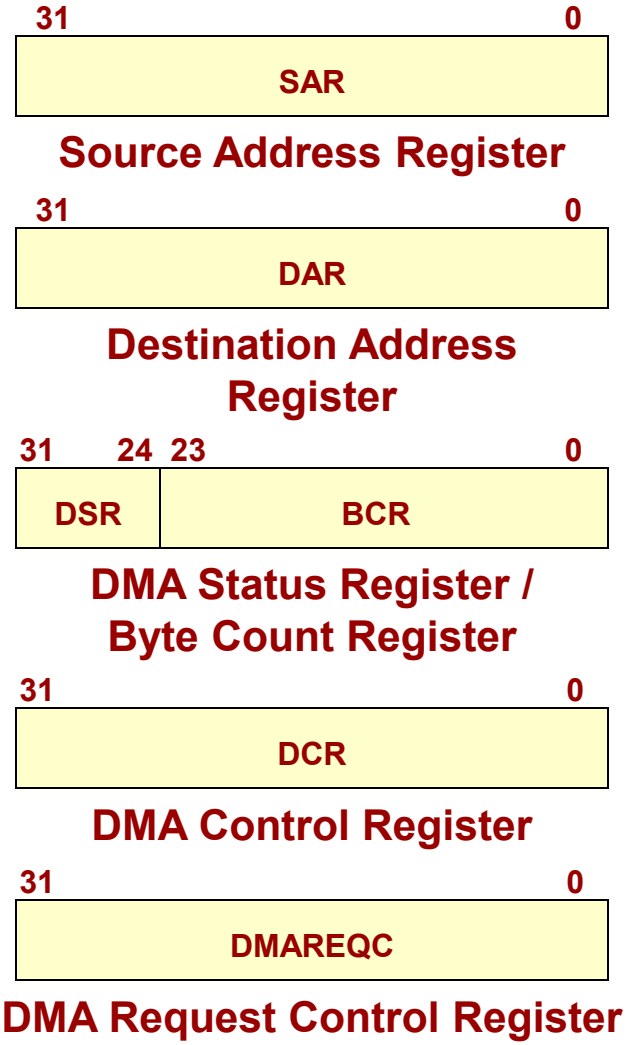
MCF5211 DMA Engines

- 4 DMA engines/channels
- Each can be configured to be started/controlled by the processor or by certain I/O peripherals
  - UARTs and special DMA timers can request a DMA operation on its behalf w/o processor help)
- 4 DMA engines and processor make requests to an arbiter who assigns control of the bus
  - Usually winning requestor has control of the bus until it relinquishes it (turns off its request signal)



MCF5211 DMA Registers

- Source / Destination Address Register hold current addresses to read from / write to
- Byte Count Register hold remaining number of bytes to be transferred
- Status Register holds information regarding error conditions and whether the channel is busy, done or being requested
- Control Register holds bits to control the operation of the DMA engine
- DMA Request Control associates requestors (DMA timer, UART Rx, UART Tx) with each of the 4 DMA channels



Control Features

- Cycle Steal or Continuous
  - Cycle steal releases the bus after each read/write
  - Continuous will not release the bus until the transfer finishes or bandwidth control unit is reached
- Bandwidth control
  - Releases the bus after a certain amount of data has been transferred (16KB, 32KB, etc.)
- Address increment or remain constant
- Data transfer size (byte, half, word, line=16 byte burst)
- Linked Control can automatically start another DMA engine when current one completes
- Note: All address and data controls can be independently set for src. and dest.