Administrivia

- Schedule
  - This is the last lecture whose material will be on Midterm 2.
  - 3/25 is a midterm review lecture.
  - 4/01 is the exam.
- 6780 projects...get started...please!

SCI Register Information & Terminology

- The information in this lecture is found:
  - Chapter 13 of the MC9S12C Family Reference manual (MC9S12C128V1.pdf) which starts at page 383.
- Break character: all logic 0s with no start, stop, or parity bits.
- Idle character: all logic 1s with no start, stop, or parity bits.
- Preamble: synchronizing idle character that begins the first transmission.

SCIBD Configuration

- SCIBD sets the baud rate.
- SCIBD is a 16 bit register, but only the bottom 13 bits are used.
- SCI baud rate = Mclk/(16 * SCIBD).
- A value of 26 (decimal) corresponds to a baud rate of 9600 for our MCU.
SCICR1 Configuration I

- Bit 0 - Parity Type (PT)
  - 0 - Even parity
  - 1 - Odd parity
- Bit 1 - Parity Enable (PE)
  - 0 - Disable parity
  - 1 - Enable parity
- Bit 2 - Idle Line Type (ILT)
  - 0 - Idle character bit count begins after start bit
  - 1 - Idle character bit count begins after stop bit
- Bit 3 - Wakeup Condition (WAKE)
  - 0 - Idle line (idle condition on RxD) wakeup
  - 1 - Address mark (1 in MSB of a received char) wakeup

SCICR1 Configuration II

- Bit 4 - Data Format (M)
  - 0 - 1 start bit, 8 data bits, 1 stop bit
  - 1 - 1 start bit, 9 data bits, 1 stop bit
- Bit 5 - Receiver Source (RSRC)
  - 0 - Internal receiver to transmitter connection
  - 1 - External receiver to transmitter connection (via the TxD pin)
- Bit 6 - SCI Stop in Wait Mode (SCISWAI)
  - 0 - SCI enabled in wait mode
  - 1 - SCI disabled in wait mode
- Bit 7 - Loop Select (LOOPS)
  - 0 - Normal operation
  - 1 - Loop operation (SCI received section is disconnected from the RxD pin allowing the RxD pin to be used for GPIO.)

SCICR2 Configuration I

- Bit 0 - Send Break (SBK)
  - 0 - No break characters
  - 1 - Transmit break characters
- Bit 1 - Receiver Wakeup (RWU)
  - 0 - Normal operation
  - 1 - Enables wakeup and inhibits receiver interrupts.
- Bit 2 - Receiver Enable (RE)
  - 0 - Disabled
  - 1 - Enabled
- Bit 3 - Transmitter Enable (TE)
  - 0 - Disabled
  - 1 - Enabled

SCICR2 Configuration II

- Bit 4 - Idle Line Interrupt Enable (ILIE)
  - 0 - IDLE interrupts disabled
  - 1 - IDLE interrupts enabled
- Bit 5 - Receiver Full Interrupt Enable (RIE)
  - 0 - RDRF and OR interrupts disabled
  - 1 - RDRF and OR interrupts enabled
- Bit 6 - Transmission Complete Interrupt Enable (TCIE)
  - 0 - TC interrupts disabled
  - 1 - TC interrupts enabled
- Bit 7 - Transmitter Interrupt Enable (TIE)
  - 0 - TDRE interrupts disabled
  - 1 - TDRE interrupts enabled
SCISR1 Configuration I

- **Bit 0 - Parity Error (PF)**
  - 0 - No parity error
  - 1 - Parity error
  - Clear PF by reading SCISR1 followed by SCIDRL. Doesn’t get set in case of OR.

- **Bit 1 - Framing Error (FE)**
  - 0 - No framing error
  - 1 - Framing error
  - Clear FE by reading SCISR1 with FE set followed by SCIDRL. Doesn’t get set in the case of OR. When sets prohibits further data reception.

- **Bit 2 - Noise Flag (NF)**
  - 0 - No noise
  - 1 - Noise
  - Clear NF by reading SCISR1 followed by SCIDRL. Doesn’t get set in the case of OR.

SCISR1 Configuration II

- **Bit 3 - Overrun (OR)**
  - 0 - No overrun
  - 1 - Overrun
  - Incoming data is lost, but the current data is intact. Clear OR by reading SCISR1 with OR set followed by SCIDRL.

- **Bit 4 - Idle Line (IDLE)**
  - 0 - Receiver input is active or has never become active since last IDLE flag clear
  - 1 - Receiver input is idle
  - Clear IDLE flag by reading SCISR1 with IDLE set followed by SCIDRL.

- **Bit 5 - Receive Data Register Full (RDRF)**
  - 0 - Data not available in SCI data register
  - 1 - Received data available in SCI data register
  - Clear RDRF by reading SCISR1 with RDRF set followed by SCIDRL.

SCISR2 Configuration

- **Bit 6 - Transmit Complete (TC)**
  - 0 - Transmission in progress
  - 1 - No transmission in progress
  - Clear TC by reading SCISR1 with TC set then writing to SCIDRL. TC is set when the TDRE flag is set and no data, preamble, or break character is being transmitted.

- **Bit 7 - Transmit Data Register Empty (TDRE)**
  - 0 - No byte transferred to the transmit shift register
  - 1 - Byte transferred to transmit shift register
  - Clear TDRE by reading SCISR1 with TDRE set followed by writing to SCIDRL.
SCIDRL & SCIDRH Configuration

- SCIDRL is used for bits 0-7 for transmit and receive.
- SCIDRH bit 6 is the ninth data bit transmitted when in 9-bit mode.
- SCIDRH bit 7 is the ninth data bit received when in 9-bit mode.
- When running in 9-bit mode access SCIDRH before SCIDRL.

SCI Initialization

- SCIBD = 26; //9600 baud
- SCICR1 = 0x00; //no parity, 8 data bits, normal operation
- SCICR2 = 0x2C; //receiver & transmitter enable, //RDRF interrupt enable

SCI Transmit Ritual

- Check for TDRE by reading SCISR1.
- Write data to SCIDRL.

if(SCISR1 & TDRE) {
    SCIDRL = data;
}

SCI Receive Ritual

- Check for RDRF by reading SCISR1.
- Read data from SCIDRL.

if(SCISR1 & RDRF) {
    data = SCIDRL;
}
SCI Initialization

```c
#define TDRE 0x80
#define RDRF 0x20
#define TXINT 0x80

void SCI_Init(void){
    asm sei
    RxFifo_Init(); // empty FIFOs
    TxFifo_Init();
    SCIBD = 52;   // 9600 bits/sec
    SCICR1 = 0;   // M=0, no parity
    SCICR2 = 0x2C; // enable, arm RDRF
    asm cli       // enable interrupts
}
```

SCI Interface ISR

```c
// RDRF set on new receive data
// TDRE set on empty transmit register
interrupt 20 void SciHandler(void){
    char data;
    if(SCISR1 & RDRF){
        RxFifo_Put(SCIDRL); // clears RDRF
    }
    if((SCICR2&TXINT)&(SCISR1&TDRE)){
        if(TxFifo_Get(&data){
            SCIDRL = data;  // clears TDRE
        }
    }
    else{
        SCICR2 = 0x2C;  // disarm TDRE
    }
}
```

SCI In/Out Character

```c
// Input ASCII character from SCI
// spin if RxFifo is empty
char SCI_InChar(void){ char letter;
    while (RxFifo_Get(&letter) == 0){};
    return(letter);
}

// Output ASCII character to SCI
// spin if TxFifo is full
void SCI_OutChar(char data){
    while (TxFifo_Put(data) == 0){};
    SCICR2 = 0xAC;  // arm TDRE
}```