Slide 1

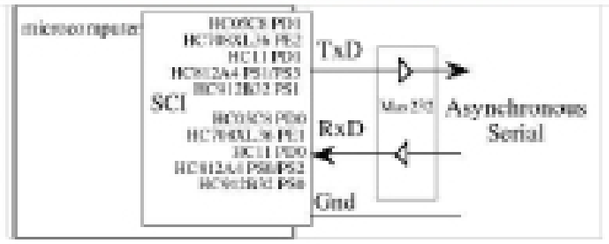
ECE/CS 3720: Embedded System Design (ECE 6960/2 and CS 6968)

Chris J. Myers

Lecture 16: Serial I/O Devices (cont)

Slide 2

SCI Hardware Interface



Slide 3

Baud Rate Selection for MC68HC11

Address: $002B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	SCP1	SCP0	SCR2	SCR1	SCR0	
Write:	TCLR							
Reset:	0	0	0	0	0	0	0	0

SCP1	SCP0	Prescaler,P
0	0	1
0	1	3
1	0	4
1	1	13

SCR2	SCR1	SCR0	Divider,BR
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Slide 4

SCI Data Register (SCDR)

Address: $002F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R7	R6	R5	R4	R3	R2	R1	R0
Write:	T7	T6	T5	T4	T3	T2	T1	T0
Reset:	Undefined by reset							

SCI Control Register 1 (SCCR1)

Address: $002C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R8	T8	0	M	WAKE	0	0	0
Write:								
Reset:	?	?	0	0	0	0	0	0

- R8 - Receive data bit 8 (used if M=1)
- T8 - Transmit data bit 8 (used if M=1)
- M - Mode, 1 creates 11-bit frame (else 10-bit frame)
- WAKE - Wake up by address mark (1) or idle (0)

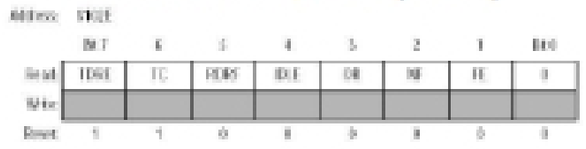
Slide 5

SCI Control Register 2 (SCCR2)

- TIE - Transmit interrupt enable
- TCIE - Transmit complete interrupt enable
- RIE - Receive interrupt enable
- ILIE - Idle line interrupt enable
- TE - Transmitter enable
- RE - Receiver enable
- RWU - Receiver wake-up control
- SBK - Send break

Slide 6

SCI Status Register (SCSR)



- TDRE - Transmit data register empty flag
- TC - Transmit complete flag
- RDRF - Receive data register full flag
- IDLE - Idle line detected flag
- OR - Overrun error flag
- NF - Noise error flag
- FE - Framing error flag

Slide 7

Output a Character Using the SCI Port

```
OutChar ldab SCSR   ;status
        bitb #$80   ;tdre?$
        beq  OutChar
        staa SCDR   ;output
        rts
OutChar staa SCDR   ;output
wait2   ldab SCSR   ;status
        bitb #$80   ;tdre?
        beq  wait2
        rts
OutChar staa SCDR   ;output
wait3   ldab SCSR   ;status
        bitb #$40   ;TC?
        beq  wait3
        rts
```

Slide 8

Output a Character Using the SCI Port in C

```
void init(void){ BAUD=0x33;  // 1200 baud
                 SCCR1=0x00; // 8data, 1stop
                 SCCR2=0x0C;}// enable gadfly
#define RDRF 0x20
#define TDRE 0x80
#define TC 0x40
unsigned char insci(void){
    while ((SCSR & RDRF) == 0);
    return(SCDR); }
void OutChar(unsigned char letter){
    while ((SCSR & TDRE) == 0);  /* Wait for TDRE */
    SCDR=letter; }               /* then output */
void outsci2(unsigned char letter){
    SCDR=letter;                 /* Output then */
    while ((SCSR & TDRE) == 0);} /* wait for TDRE */
void outsci3(unsigned char letter){
    SCDR=letter;                 /* Output then */
    while ((SCSR & TC) == 0); }  /* wait for TC */
```

Slide 9

SCI Receive Only Interrupt Interface

Bit	SCCR2
7 0	TIE=0
6 0	TCIE=0
5 1	RIE=1
4 0	ILIE=0
3 1	TE enable TxD
2 1	RE enable RxD
1 0	RWU no wake up
0 0	SBK no break

Slide 10

Ritual to Initialize to Accept Receive Interrupts

```
RITSCI sei        ;make atomic
       ldaa #$31  ;4800 baud
       staa BAUD
       ldaa #00   ;M=0, 8 bit data
       staa SCCR1 ;1 stop
       ldaa #$2C
       staa SCCR2
       bsr  CLRQ  ;Initialize FIFO
       cli        ;Enable
       rts
```


Slide 11

ISR for Receiver Interrupts

```
SCIHAN ldaa SCSR
       anda #$20  RDRF set?
       bne  INSCI
INSCI  ldaa SCSR ;status
       anda #$0E ;OR, NF, FE
       bne  ERROR
       ldaa SCDR ;data, ack
       bsr  PutFifo ;Communicate
       bcs  ERROR ;FIFO full?
       rti
```

Slide 12

SCI Receive Interrupt Interface in C

```
#pragma interrupt_handler SCIHAN()
#define RDRF 0x20
// Executed on RDRF (not TDRE)
void SCIhandler(void){
    if((SCSR&0x21)!= RDRF) error();
    if(SCSR&0x0E) error();
    if(PutFifo(SCDR)) error(); }
void RitualSCI(void){
asm(" sei");
     BAUD=0x31;    // 4800 bits/sec
     SCCR1=0;      // 8 bit 1 stop
     SCCR2=0x2C;   // Receiver intrpt
     CLRQ();       // Clear FIFO
asm(" cli");}
```