

CS8900a problems (very long post)

Source: <http://coding.derkeiler.com/Archive/General/comp.arch.embedded/2004-05/0124.html>

From: Kiran Vinta (*ghettoman101_at_yahoo.com*)

Date: 05/03/04

Date: 2 May 2004 21:21:53 -0700

Currently, I am trying to interface the a cs8900a to the motorola 6812 microcontroller. A description and schematic of the board I am using can be found at <http://www.invector.nu/iet8900.asp>. Additionally, I have been using the cs8900a in 8 bit mode (or trying to). I am able to read from registers or any place in internal memory using the packetpage pointer and packetpage data registers (i.e. I can read the product ID register). My main problem is that I I am unable to transmit any packets. I am connecting my ethernet module to my pc via a crossover cable. On my pc, I have an ethereal packet sniffer running to see if I was able to send anything. Right now I am able to get a bid for transmit (the RDY4txNow bit goes high) but it will not transmit anything. Also when I put the while loop in for the repeated transmit, the cs8900a will not reset and the link light will not come on. If I comment out the while loop, the reset light does come on but it still does transmit anything. I think it either may be compiler problems or timing issues. How long must I wait after I reset to start transmitting anything? I've looked in cirrus's 8-bit application note for transmitting and I've read the data sheet for transmitting many times (section 5.7). Im really stumped so any help will be appreciated.

Motorola ports – IET 8900a pins

PORTT[7:0] – AD[7:0]

PORTJ[7:4] – D[3:0]

PORTJ[0] – AEN

PORTJ[1] – IOR

PORTJ[2] – IOW

```
#include "HC12.h" //port maps
```

```
#define PPPtr 0x0a //address of packet page pointer
#define PPData 0x0c //address of packet page data register
#define RxTxData 0x00 // Receive/Transmit data (port 0)
#define RxTxData1 0x02 // Receive/Transmit data (port 1)
#define TxCmd 0x04 // Transmit Command
#define TxLength 0x06 // Transmit Length
#define ISQ 0x08 // Interrupt status queue
```

```

#define bkpt asm("bgnd");

//waits a time/4 ms
void mswait(unsigned short time){

    for(;time>0;time--){

        TC5=TCNT+2000; // wait

        TFLG1 = 0x20; // clear C5F

        while((TFLG1&0x20)==0){};
    }
}

void TimerInit(void){

    COPCTL = 0x00; // disable COP
    TIOS |= 0x20; // enable OC5
    TSCR =0x80; // TEN(enable)

}

//changing the wait periods does not change the final data output
unsigned char IOread (unsigned char address)
{
    unsigned char data;
    DDRT = 0x00; //set port t as input
    PORTJ = ((address<<4)&0xF0) | 0x07; //put address on pins
    PORTJ &= ~0x01; //pulsing AEN
    PORTJ &= ~0x02; //IOR
    mswait(1); //wait
    data = PORTT; //capture data
    PORTJ |=0x02; //end pulse IOR
    PORTJ |= 0x01; //end pulse AEN
    return data;
}

//changing the wait periods does not change the final data output
void IOwrite (unsigned char address, unsigned char value)
{
    DDRT = 0xff; //set port as output
    PORTT = value; //Port T equals data value
    PORTJ = ((address<<4)&0xF0) | 0x07; //put address on pins
    PORTJ &= ~0x01; //pulsing AEN
    PORTJ &= ~0x04; //IOwrite
    PORTJ &= ~0x02;
    mswait(1);
    PORTJ |=0x02;
    PORTJ |=0x04; //end IOW
    PORTJ |= 0x01; //end pulse AEN
}

```

```

}

unsigned char reading; //first byte for input
unsigned char reading1; //second byte for input
unsigned char BusST0, BusST1;
unsigned char event0, event1;

void main(void){
  DDRH = 0xFF; //set Port H as outputs
  event0 = 0x00;
  reading = 0x00; //setting reading to zero
  reading1 = 0x00; //setting reading1 to zero
  COPCTL = 0x00; //disable COP
  // SCI_Init(13); //initialize SCI interface
  TimerInit(); //initialize timer unit
  DDRJ = 0xFF; //set port j as outputs
  PORTJ |= 0x07; //
  PORTH |= 0x01; //pulse reset
  mswait(1000); //250 ms
  PORTH &= ~0x01; //end pulse reset
  // mswait(1);

  // Configure RxCTL fo Promiscious mode, RxOK
  // (1) Write 0x0104 to PacketPage Pointer
  // (2) Write 0x0180 to PacketPage Data Port
  IOWrite(PPPptr, 0x04);
  IOWrite(PPPptr + 1, 0x01);
  IOWrite(PPData, 0x80);
  IOWrite(PPData + 1, 0x01);

  // Set 10BaseT, SerRxOn, SerTxOn in LineCTL
  // (1) Write 0x0112 to PacketPage Pointer
  // (2) Write 0x00c0 to PacketPage Data Port
  IOWrite(PPPptr, 0x12);
  IOWrite(PPPptr + 1, 0x01);
  IOWrite(PPData, 0xc0);
  IOWrite(PPData + 1, 0x00);

  while (1)
  {
    // Send the transmit command
    IOWrite(TxCmd, 0xc0); //wait for whoel fram
    IOWrite(TxCmd + 1, 0x00);

    // 8 bytes to be sent
    IOWrite(TxLength, 0x04); //sending 8 byte
    IOWrite(TxLength + 1, 0x00);
  }
}

```

comp.arch.embedded: CS8900a problems (very long post)

```
IOWrite(PPPTr, 0x38); //address of busST register
IOWrite(PPPTr + 1, 0x01);

// Bid for transmit
do {
    BusST0 = IOread(PPData);
    BusST1 = IOread(PPData+1);
}while(!(BusST1==0x01)); //check for rdy4txnow
```

```
IOWrite(RxTxData, 0x12); //sending test data
IOWrite(RxTxData+1, 0x23);
IOWrite(RxTxData, 0x45);
IOWrite(RxTxData+1, 0x67);
IOWrite(RxTxData, 0x89);
IOWrite(RxTxData+1, 0xab);
IOWrite(RxTxData, 0xcd);
IOWrite(RxTxData+1, 0xef);
```

```
}
```

```
}
```

–Kiran Vinta

ps Ive talked to the invector but they are very slow in responding.