Jonathan W. Valvano February 25, 2000, 11:00am-11:50am

(45) Question 1. In this problem you will design a period meter using interrupting input capture.

(10) Part a) Δp•65536>100ms, or Δp>1.5μs, so chose Δp=2μs.

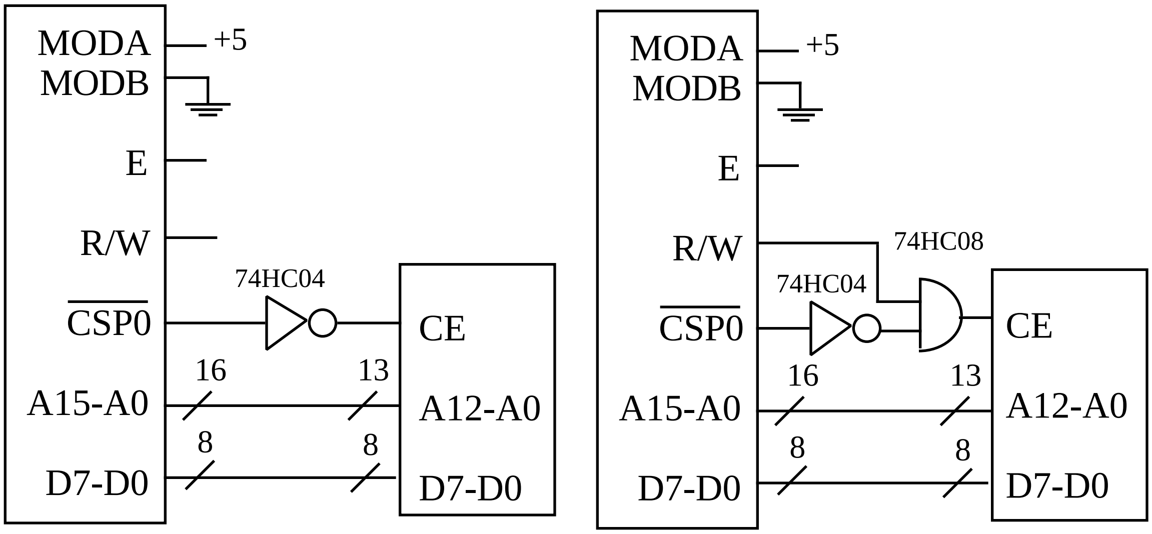
(5) Part b) The fixed-point number system has Δ=2μs. 12.346789 ms is stored as
12.346789ms/0.002ms = 6173

(30) Part b) Modify this program from Chapter 6

```
// PT7/IC7  input = external signal
// rising edge to rising edge
// resolution = 2us
// Range = 100 μs to 100 ms   ,
// no overflow checking
// IC7  interrupt each period,
unsigned int Period;  // units of 2us
unsigned int First;   // TCNT first edge
unsigned char Done;   // Set each rising
void Ritual(void){
    asm(" sei");    // make atomic
    TIOS &= 0x7F ;   // PT7  input capture
    DDRT &= 0x7F ;   // PT7  is input
    TSCR = 0x80;    // enable TCNT
    TMSK2= 0x34 ;    // 2us  clock
    TCTL3 = (TCTL3&0x3F)|0x40     ; // rising
    First = TCNT;   // first will be wrong
    Done=0;         // set on subsequent
    TFLG1 = 0x80 ;   // Clear C7F
    TMSK1 |= 0x80 ;  // Arm IC7
    asm(" cli");}
#pragma interrupt_handler TIC7handler  ()
void TIC7handler  (void){
    Period=TC7 -First;
    First=TC7 ;   // Setup for next
    TFLG1=0x80 ;  // ack by clearing C7F
    Done=0xFF;}
#pragma abs_address:0xffe0
void (*TC7_vector[])() = { TIC7handler};
#pragma end_abs_address
```

(45) Question 3. In this problem you will interface an 8192 byte ROM to the MC68HC812A4

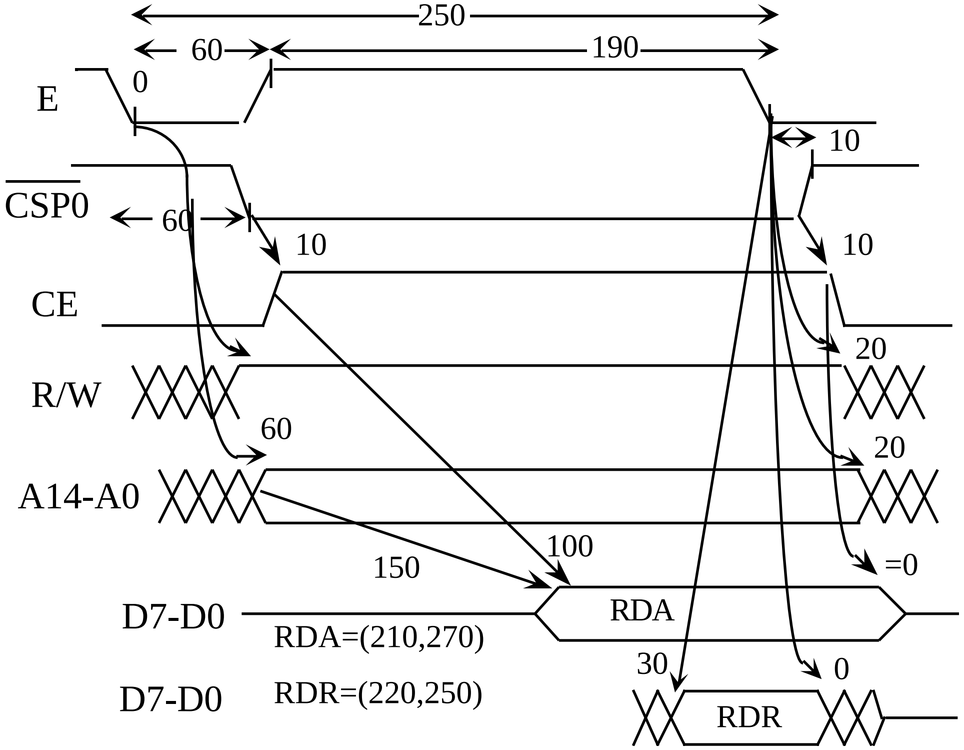
(15) Part a) There are two good answers. Remember CSP0 is a clock signal.



(15) Part b) RDA = (later (60+150, ↑CE+ 100), earlier (AdN, ↓CE))
= (later (210, 60+10+100), earlier (t_1+20, t_1+10+10))
= (210, t_1+20)

RDR = (t_1 - 30 , t_1)

To make RDA overlap RDR, 210 ≤ t_1 - 30. One cycle stretch (t_1=250) will make 240 ≤ t_1
(15) Part c) Show the combined **read cycle** timing diagram.



(10) Question 2.

(5) Part a) This function is **not reentrant**. It has two nonatomic write-read sequences

```
 F03F  1805020800        movw 2,x,_r  ; r=x;
 F044  FC0800            ldd _r       ; RegD=r
```

and

```
 F049  7C0800            std _r       ; r=r+y
 F04C  FC0800            ldd _r
```

It has one nonatomic read-modify-write sequence

```
 F044  FC0800            ldd _r       ; RegD=r
 F047  E306              addd 6,x     ; RegD=r+y
 F049  7C0800            std _r       ; r=r+y
```

(5) Part b) This function is **reentrant**. It has no nonatomic write-read, read-modify-write, or write-write sequences. The only access to a global is a single atomic write.