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

(a) $\Delta p \cdot 65536 > 100\text{ms}$, or $\Delta p > 1.5\mu\text{s}$, so choose $\Delta p = 2\mu\text{s}$.

(b) The fixed-point number system has $\Delta = 2\mu\text{s}$. 12.346789 ms is stored as 12.346789 ms ÷ 0.002 ms = 6173

Modify this program from Chapter 6

```
void Ritual(void){
  asm(" sei");    // make atomic
  TIOS &= 0x7F;   // PT7 input capture
  DDRT &= 0x7F;   // PT7 is input
  TSLA = 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(){
  Period = TC7 - First;
  First = TC7;  // Set up for next
  TFLG1 = 0x80;  // ack by clearing C7F
  Done = 0xFF;}
```

```
# pragma abs_address: 0xFFE0
void (*TC7_vector[])() = { TIC7handler};
```

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

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

(b) $RDA = (\text{ later (60+150, CE+100), earlier (AdN, CE)})$

$RDA = (\text{ later (210, 60+10+100), earlier (t_1+20, t_1+10+10)})$

$RDA = (210, t_1+20)$

$RDR = (t_1 - 30, t_1)$
To make RDA overlap RDR, \(210 \leq t_1 - 30\). One cycle stretch \((t_1=250)\) will make \(240 \leq t_1\)

(15) Part c) Show the combined **read cycle** timing diagram.

---

**Diagram:**

- **RDA** (Read Data Access) starts at \(210\) and ends at \(270\).
- **RDR** (Read Data Return) starts at \(220\) and ends at \(250\).
- Arrows indicate the timing of the read operations.

---

**Question 2.**

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

\[
\begin{align*}
F03F & \quad 1805020800 & \quad \text{movw} \ z, x, _r & \quad ; r=x; \\
F04C & \quad FC0800 & \quad \text{ldd} \ _r & \quad ; \text{RegD}=r
\end{align*}
\]

and

\[
\begin{align*}
F049 & \quad 7C0800 & \quad \text{std} \ _r & \quad ; r=r+y \\
F04C & \quad FC0800 & \quad \text{ldd} \ _r & \quad
\end{align*}
\]

It has one nonatomic read-modify-write sequence

\[
\begin{align*}
F044 & \quad FC0800 & \quad \text{ldd} \ _r & \quad ; \text{RegD}=r \\
F047 & \quad E306 & \quad \text{add} \ d, x & \quad ; \text{RegD}=r+y \\
F049 & \quad 7C0800 & \quad \text{std} \ _r & \quad ; r=r+y
\end{align*}
\]

(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.