Recap
Pointers
Indexed mode
Arrays and strings

Overview
Timer
Fixed-time delay
Debugging
Intrusiveness
Monitors and dumps

4.5. 16-bit timer

$$0044 \quad b_{15} \quad b_{14} \quad b_{13} \quad b_{12} \quad b_{11} \quad b_{10} \quad b_{9} \quad b_{8} \quad b_{7} \quad b_{6} \quad b_{5} \quad b_{4} \quad b_{3} \quad b_{2} \quad b_{1} \quad b_{0}$$

<table>
<thead>
<tr>
<th>Addr</th>
<th>Bit 7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>Bit 0</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0044</td>
<td>Bit 15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>Bit 8</td>
<td>TCNT</td>
</tr>
<tr>
<td>$0045</td>
<td>Bit 7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>Bit 0</td>
<td>TCNT</td>
</tr>
<tr>
<td>$0046</td>
<td>TEN</td>
<td>TSWAI</td>
<td>TSFRZ</td>
<td>TFFCA</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>TSCR1</td>
</tr>
<tr>
<td>$004D</td>
<td>TOI</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>TCRE</td>
<td>PR2</td>
<td>PR1</td>
<td>PR0</td>
</tr>
<tr>
<td>$004F</td>
<td>TOF</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>TFLG2</td>
</tr>
</tbody>
</table>

Table 4.11. 9S12 timer ports.

<table>
<thead>
<tr>
<th>PR2</th>
<th>PR1</th>
<th>PR0</th>
<th>Divide by</th>
<th>E = 8 MHz</th>
<th>E = 24 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TCNT period</td>
<td>TCNT frequency</td>
</tr>
<tr>
<td>PR2</td>
<td>PR1</td>
<td>PR0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>125 ns</td>
<td>8 MHz</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>250 ns</td>
<td>4 MHz</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>500 ns</td>
<td>2 MHz</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>1 µs</td>
<td>1 MHz</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>16</td>
<td>2 µs</td>
<td>500 kHz</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>32</td>
<td>4 µs</td>
<td>250 kHz</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>64</td>
<td>8 µs</td>
<td>125 kHz</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>128</td>
<td>16 µs</td>
<td>62.5 kHz</td>
</tr>
</tbody>
</table>

Table 4.12. Given an E clock frequency, the PR2 PR1 and PR0 bits define the TCNT rate.

Fixed time delay software using the built-in timer

void Timer_Init(void){
    TSCR1 = 0x80; // enable TCNT
    TSCR2 = 0x00; // 125ns
}
void Timer_Wait(unsigned short cycles){
    unsigned short startTime = TCNT;
    while((TCNT-startTime) <= cycles){}
}

****Timer_Init*******
* Initialize Timer
* Input: none
* Outputs: none
* error: none

Timer_Init
    movb #$80,TSCR1    enable TCNT
    movb #$00,TSCR2    ;125ns in RUN mode
    rts

Jonathan W. Valvano
Introduction to Embedded Microcomputer Systems  Lecture 12.2

;***Timer_Wait**********
; Time delay function
; Input: RegD time to wait (125ns)
; Outputs: none
; error: input must be less than 60000
Timer_Wait
    std delay ;time to wait
    movw TCNT,start ;TCNT at start
Wloop ldd TCNT       ;now
    subd start ;soFar=TCNT-Start
    cpd delay
    Wchk bl0 Wloop ;loop if soFar<delay
    rts

Show Timer_Wait starter file
    Run until start is 50000  RegD is about 14000
    Put a scan point at Wchk
    Watch TCNT roll over
    It works because all data are 16-bit unsigned

Real time systems
    Bounded latency
    Input interface:
        input interface latency RDRF -> Read SCIDRL
    Output interface:
        output interface latency TDRE -> Write SCIDRL
    Periodic process (square wave, Labs 7, 8 and 9):
        software activity occurs at a periodic rate, $\Delta t$
        let $t_n$ be the nth time the process executes
        goal to make $t_n - t_{n-1} = \Delta t$
        $\delta t = \text{jitter}$
        $\Delta t - \delta t < t_i - t_{i-1} < \Delta t + \delta t$ for all $i$

*******************TimerWait************************
Does the LED flash at exactly 5 Hz?
    Put a ScanPoint at PTP toggle, measure jitter

show the sliding time
    measure the jitter
    $100\text{ms}-\delta t < t_n - t_{n-1} < 100\text{ms}+\delta t$
fix to have almost perfect timing
    measure the jitter again

Intrusiveness
    degree of perturbation caused by the debugging itself
    how much the debugging slows down execution
Nonintrusive
    characteristic or quality of a debugger
    allows system to operate as if debugger did not exist
    e.g., logic analyzer, ICE, BDM
Minimally intrusive
    negligible effect on the system being debugged
    e.g., dumps (ScanPoint) and monitors
Highly intrusive
    e.g., print statements, breakpoints and single-stepping
Dump or ScanPoint

Question: Is my program outputting to the motor?
Method: Add instruments, assemble, download, run

Initialization
dbuf rmb 100 first 100 outputs to stepper

dpt rmb 2 pointer to next place

LDX #dbuf
STX dpt
LDAA #100

DLOOP CLR 1,x+

DBNE A,DLOOP

OR

LDY #dbuf
STY dpt

DLOOP CLR 1,y+

CPY #dbuf+100
BNE DLOOP

Debugging Instrument

ORG *

dscan pshx
LDX dpt
CPX #dbuf+100
BHS DDONE

MOVB PTT,1,x+
STX dpt
PULX

DDONE RTS
NOP

DSIZE EQU *-DSCAN

Is it Intrusive??

1) count cycles

$F05E ORG *
$F05E 34 [2]( 0) DSCAN PSHX
$F05F FE0800 [3]( 2) LDX DPT
$F062 8E0866 [2]( 5) CPX #dbuf+100
$F065 2409 [3]( 7) BHS DDONE
$F067 1809300024 [5]( 10) MOVB PTT,1,x+
$F06C 7E0800 [3]( 15) STX DPT
$F06F 30 [3]( 18) PULX
$F070 3D [5]( 21) DDONE RTS
$F071 A7 [1]( 26) NOP
$0014 DSIZE EQU *-DSCAN

2) TCNT measurement

TEST1 MOVB #$80,TSCR1
MOVW TCNT,FIRST
JSR DSCAN
LDD TCNT
SUBD FIRST
STD DELAY

3) Attach scope to unused output, PT7

TEST2 BSET DDRT,#$80
TLOOP LDX #dbuf
STX DPT
BSET PTT,#$80
JSR DSCAN
BCLR PTT,#$80
BRA TLOOP
LED monitor (PP7)

Question: Is my program running?
Method: Add instruments, assemble, download, run

Initialization

\[
\text{DDRP} \leftarrow 0x80; \\
\text{bset DDRP,} #80
\]

Debugging Instrument (toggle PP7)

\[
\text{PTP} \leftarrow 0x80; \\
\text{ldaa PTP} \ ; [3] \text{ cycles} \\
\text{eora} \ #80 \ ; [1] \text{ cycle} \\
\text{staa PTP} \ ; [3] \text{ cycles}
\]

How intrusive is the heartbeat?

The bottom line

- Timer is an accurate wait to create delays
- Monitor is a real-time visualization
- Dump records data for later analysis
- Simulation (testing), prototype (test), final system (test)