Timers
Planning

• Theory
  - What is a timer?

• Usage
  - Examples

• Electronics
  - How does it work?

• HC11
  - Basic usage
  - Programming
Definition

• What is a timer?
  - A timer is an oscillator that beats at precise and programmable frequency.

• Built-in timer (like in 68HC11)
  - The frequency is linked to the one of the IC

• External timer
  - Larger ranges of frequencies
Case of 68HC11

- Built-in timer
  - Activated by registers
  - Quite limited
- At what frequency does it beat?
  - We can use 4 frequency: E-clock, one quarter, one eighth or one sixteenth.
Case of 68HC11

• The main counter
  - The counter begins at $0000$ and goes up to $FFFF$ (16 bits counter).
  - When it reaches $FFFF$, it sets up an overflow-flag.
  - Many other registers are used
Other timers

- Specific electronic components
  - Implemented outside the IC
  - The frequency is not linked to the IC one
- Three main operating modes of timers
  - The monostable multivibrator
  - The bistable multivibrator
  - The astable multivibrator
Monostable

- Hold a given state as it is powered (high)
- Switches to an unstable state (low) for a given duration
  - From $\mu$s to hours
  - Not parametrable

\[ \Delta t \]

Input signal

High

Low
Bistable

- Hold either the high or low state
- Switches from on to the other when an input trigger is connected
  - Quartz
  - Sensors
- Not very useful
Astable

- Automatically switches between its high and low states
  - Rectangle wave generator
- If the low state time equals the high one
  - Square wave generator
Frequency

<table>
<thead>
<tr>
<th>Timer</th>
<th>Frequency</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>555 Timer (M&amp;A)</td>
<td>1.1 ms</td>
<td>Most IC ever built</td>
</tr>
<tr>
<td>LM567</td>
<td>2 μs</td>
<td>100 s</td>
</tr>
<tr>
<td>MM5369</td>
<td>250 ns</td>
<td>32 ms</td>
</tr>
</tbody>
</table>

• Multistages timers (MM5369: 17 stages)
Price of timers

- Very cheap IC
  - 555 timer: $0.50
  - MM5369: $4.50
- Easy to provide
- Easy to use
- Robust components
Usage of timers

• Where can timers be found?
  - EVERYWHERE!
  - Clocks
  - Digital camera
  - Radar
  - Space exploration
  - Anywhere you could think to put one!
  - Anything that could be linked to time
Examples - Input

- Time between two rising edges
  - Radar

- Compute the time between two successive falling edges
  - Track & Field Timing

- Pulse width measurement
  - Time delay
Examples - Output

- **Time basis**
  - Clocks
  - Rectangle wave generator
    - Use specific registers (see HC11 specific part)

- **Creating a time delay**
  - 10 ms delay to program an EEPROM
    - 10 ms at 2MHz $\Leftrightarrow$ $4E20$ cycles (20,000)
How to use it?

- **Software solution (HC11 case)**
  - Implement register
  - Read registers
- **Hardware solution**
  - Often the timer can not be easily changed
  - Counter is a specific added part
• Composition of timers

- Oscillator
  - Crystal
- Logical components
  - Analogic circuitry
Important dates

- 1880 Piezoelectric effect discovered by Jacques and Pierre Curie
- 1905 First hydrothermal growth of quartz in a laboratory - by G. Spezia
- 1918 First use of piezoelectric crystal in an oscillator
- 1927 First quartz crystal clock built
- 1934 First practical temp. compensated cut, the AT-cut, developed
- 1956 First commercially grown cultured quartz available
Piezzo – electric effect

- “pressure-electric” : piezein = to press, in Greek

Undeformed lattice

Strained lattice
Different couplings

- Fundamental Mode: Thickness Shear
- Third Overtone: Thickness Shear
- Flexure Mode
- Thickness Shear Mode
- Extensional Mode
- Face Shear Mode

<table>
<thead>
<tr>
<th>STRAIN</th>
<th>FIELD along:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x</td>
</tr>
<tr>
<td>EXTENSIONAL</td>
<td>x</td>
</tr>
<tr>
<td>along:</td>
<td>y</td>
</tr>
<tr>
<td></td>
<td>z</td>
</tr>
<tr>
<td>SHEAR about:</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>y</td>
</tr>
<tr>
<td></td>
<td>z</td>
</tr>
</tbody>
</table>
The resonator

• Composition

Metallic electrodes

Resonator plate substrate (the “blank”)
The resonator

• Electronic symbol

• Equivalent circuit
The oscillator

- Pierce
- Colpitts
- Clapp
- Butler
- Modified Butler
- Gate
The timer

- A very common one: the NE555
The timer

Timers

Introduction
Timers?
Usage
Electronics
HC11
Conclusion

Timers

HC11
Usage Electronics

The timer

Timers

Introduction
Timers?
Usage
Electronics
HC11
Conclusion

Timers

HC11
Usage Electronics
Timers

The timer

- A simple example of use:

  Tune the on/off ratio of oscillations
The timer

HC11

Usage

Fig. 3
Electronic circuitry

- 20 transistors
- 15 resistors
- 2 diodes
HC11 timer
General description

- **Main Timer**
  - 16-bit free running counter with a prescaler. Readable but not writable during operation
  - Four functions:
    - Periodic interrupts: real-time interrupt (RTI)
    - Input capture
    - Output compare
    - Pulse accumulator
  - Main Timer System Block Diagram

<table>
<thead>
<tr>
<th>Bit 15</th>
<th>Bit 14</th>
<th>Bit 13</th>
<th>Bit 12</th>
<th>Bit 11</th>
<th>Bit 10</th>
<th>Bit 9</th>
<th>Bit 8</th>
<th>$100E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Bit 7</td>
<td>Bit 6</td>
<td>$100F</td>
</tr>
<tr>
<td>Bit 7</td>
<td>Bit 6</td>
<td>Bit 5</td>
<td>Bit 4</td>
<td>Bit 3</td>
<td>Bit 2</td>
<td>Bit 1</td>
<td>Bit 0</td>
<td></td>
</tr>
</tbody>
</table>
General description

• Port A pin outs
  - PA0~PA2, input-only, serves as IC3~IC1
  - PA3, bidirectional, serves as:
    • IC4 or OC5 (determined by PACTL I4/O5 bit)
    • or general-purpose I/O (determined by PACTL DDRA3 bit)
  - PA4~PA6, output-only, serves as OC4~OC2
  - PA7, bidirectional, serves as:
    • input to pulse accumulator, special OC1,
    • or general-purpose I/O (determined by DDRA7)
Timers

Counter - Prescaler

- **Prescaler:**
  - Selection of 4 clocking rates by setting PR0, PR1 in TMSK2
  - Must be done in the first 64 bus cycles
  - Trade-off between timer resolution and timer range
  - Overflows increase program complexity

- A look at Clock divider chains
Major Clock Divider Chains
Counter - prescaler

• Timer Overflow
  - Timer overflow flag (TOF) status bit set each time the counter rolls over from $FFFF$ to $0000$
  - TOF status bit can generate an automatic interrupt request if the timer overflow interrupt (TOI) enable bit is set to 1
  - Registers associated: TMSK2, TFLG2
Timers

Clearing timer flags
- Load an accumulator with a mask that has a one in the bit(s) corresponding to the flag(s) to be cleared
- Then write this value to TFLG1 or TFLG2 to clear flags
  • E.g. LDAA #$80
  STAA TFLG2
  STAA TFLG2 (Note: TFLG2 has been equated to address $1025)
- Or use BCLR instruction to clear the flag, the mask should have ones in the bit positions corresponding to the flags to be cleared and zeros in all other bits. (BCLR read->AND with inversed mask->write back)
  • E.g. LDX #TFLG2
  BCLR $00,X #10000000
Counter - prescaler

• Caution!
  - Don’t use BSET to clear flags
    • Because it could inadvertently clear one or more of other flags in register
Real-Time interrupt

- Generates h/w interrupts at fixed periodic rates.
- Used for longer delays (or short loop delays)
- Associated registers: TMSK2, TFLG2, PACTL

<table>
<thead>
<tr>
<th>TMSK2</th>
<th>RTII</th>
<th>$1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFLG2</td>
<td>RTIF</td>
<td>$1025</td>
</tr>
<tr>
<td>PACTL</td>
<td>RTR1 RTR0</td>
<td>$1026</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RTR1</th>
<th>RTR0</th>
<th>2^13/E Divided by</th>
<th>RTI Clock Cycle (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4.10</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>8.19</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>4</td>
<td>16.38</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>8</td>
<td>32.77</td>
</tr>
</tbody>
</table>
Real-Time interrupt

- Steps to generate a real-time h/w interrupt:
  - Select prescalar factor using bits RTR1, RTR0
  - Enable RTII bit in TMSK2
  - Clear RTIF by writing a one to it
  - After interrupt request has occurred and ISR is executed, RTIF is cleared again
Input Capture

- Used to record time external event occurs.
- Accomplished by capturing content of free-running counter when selected edge is detected at particular timer input pin.
- Timer counter content is saved in Input Capture register
- TICx registers are not affected by reset and cannot be written by s/w.
Input Capture

- Operate independently of each other.
- Read of high-order byte of TIC register inhibits new capture transfer for one bus cycle - to make sure captured two bytes are stored in appropriate address/register
- Inhibited capture will be delayed but will not be lost
# Input Capture

**Associated registers:** TIC1~TIC3, TI4/O5, TMSK1, TFLG1, TCTL2, PACTL (for IC4, clear DDRA3, set I4/O5)

<table>
<thead>
<tr>
<th>TIC1~TIC3</th>
<th>b7</th>
<th>b0</th>
<th>$</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$1010, 1012, 1014</td>
<td>Capture disabled</td>
</tr>
<tr>
<td>TFLG1</td>
<td>OC1F</td>
<td>OC2F</td>
<td>OC3F</td>
<td>OC4F</td>
</tr>
<tr>
<td>TMSK1</td>
<td>OC1I</td>
<td>OC2I</td>
<td>OC3I</td>
<td>OC4I</td>
</tr>
<tr>
<td>TCTL2</td>
<td>EDG4B</td>
<td>EDG4A</td>
<td>EDG 1B</td>
<td>EDG 1A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EDGxB</th>
<th>EDGxA</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Capture disabled</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Capture on rising edges only</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Capture on falling edges only</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Capture on any edge</td>
</tr>
</tbody>
</table>
Timer Input Capture 4/Output Compare 5 Register

Use TI4/O5 as either an input capture register or an output compare register, depending on the function chosen for the PA3 pin. To enable it as an input capture pin, set the I4/O5 bit in the pulse accumulator control register (PACTL) to logic level 1. To use it as an output compare register, set the I4/O5 bit to a logic level 0.

Register name: Timer Input Capture 4/Output Compare 5 (High)  Address: $101E

<table>
<thead>
<tr>
<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>
</tr>
</thead>
<tbody>
<tr>
<td>Read:</td>
<td>Bit 15</td>
<td>Bit 14</td>
<td>Bit 13</td>
<td>Bit 12</td>
<td>Bit 11</td>
<td>Bit 10</td>
<td>Bit 9</td>
</tr>
<tr>
<td>Write:</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Reset: 1 1 1 1 1 1 1 1

Register name: Timer Input Capture 4/Output Compare 5 (Low)  Address: $101F

<table>
<thead>
<tr>
<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>
</tr>
</thead>
<tbody>
<tr>
<td>Read:</td>
<td>Bit 7</td>
<td>Bit 6</td>
<td>Bit 5</td>
<td>Bit 4</td>
<td>Bit 3</td>
<td>Bit 2</td>
<td>Bit 1</td>
</tr>
<tr>
<td>Write:</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Reset: 1 1 1 1 1 1 1 1

Timer Input Capture 4/Output Compare 5 Register Pair (TI4/O5)
Input Capture

- Used to measure period/frequency (capture successive edges with same polarity) of signal

- Measure pulse width. (capture successive edges with alternate polarity, i.e., Rising and Falling edges)

- As time reference (used in conjunction with an OC function.)
Input Capture

- Measure as short as period of one timer count by connecting signal to two IC pins. Theoretically.

- Measuring periods longer than counter range by counting overflows

- An example program
Sample program: Using interrupts measure pulse width

REGBAS EQU $1000  *starting address of register block
PVIC1 EQU $00EB  *EVB pseudo vector for ICI
TIC1 EQU $510  *ICI register
TCL2 EQU $21  *EDG4B,EDG4A,EDG1B,EDG1A,EDG3A
SMK1 EQU $22  *IC1...I4/O51,ICI,...,IC1F
TFLG1 EQU $23  *IC1F...I4/O5F,IC1F,...,IC1F

ORG $1040  *s/w mode flag:ff-off,0-1st,1-last edge
IC1MOD RMB 1  *flag:0-not done,1-pulse measured
IC1DUN RMB 1  *time of first edge(16-bits)
FSTDG RMB 2  *pulse width(16-bits)

FW RMB 2

*initialization section
LDAA #$7E  *JMP (extented) opcode
STAA PVIC1  *EVB’s ICI pseudo vector
LDX $SSRC1  *address of ICI service routine
STX PVIC1+1  *finish JMP instruction to ICI routine

*Main section of pulse width measurement
TOP LDX REGBAS  *point to register block
LDAA #$00000000
STAA TCL2,X  *EDG1B:EDG1A=0:1,detect I1 rising edge
LDAA #$FF
STAA IC1MOD  *FF=ICI off;
CLR IC1DUN  *signal pulse not done
BCLR TLC1,X $FB  *clear old ICI flag(if any)
BSET TMSK1,X $04  *enable ICI interrupts
CLI  *enable interrupts

WAIT LDAA IC1DUN  *sets after pulse done
BEQ WAIT  *loop till pulse has been timed
SEI  *pulse done, disable interrupts

....  *display pulse width, etc.
BRA TOP  *go to top of main & repeat

*SRIC1-ICI service routine
SRIC1 LDX REGBAS  *point to register block
INC IC1MOD  *FF->0 @ 1st edge, 0->1 @ 2nd
BNE NO1ST  *if not 0, this is trailing edge

*process leading edge of pulse
LDD TIC1,X  *read time of first edge
STD FSTDG  *save till next capture
BRA O2IC1  *reconfig ICI for falling edge

BCLR TLC1,X $30  *EDG1B:EDG1A->0:0
BSET TLC1,X $20  *EDG1B:EDG1A->1:0

*process trailing edge of pulse
NO1ST LDD TIC1,X  *get time of trailing edge
SURD FSTDG  *time of 2nd minus time of 1st
STD PW  *store pulse width in hex number cycles.
BCLR TLC1,X $30  *disable ICI
LDAA #1
STAA IC1DUN  *signal pulse measured
BCLR TFLG1,X $FB  *clear IC1F
RTI  *return from ICI service
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Type</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGBAS</td>
<td>EQU</td>
<td>$1000</td>
<td>Starting address of register block</td>
</tr>
<tr>
<td>PVIC1</td>
<td>EQU</td>
<td>$00E8</td>
<td>EVB pseudo vector for IC1</td>
</tr>
<tr>
<td>TIC1</td>
<td>EQU</td>
<td>$10</td>
<td>IC1 register</td>
</tr>
<tr>
<td>TCTL2</td>
<td>EQU</td>
<td>$21</td>
<td>EDG4B, EDG4A, EDG1B, EDG1A, ..., EDG3A</td>
</tr>
<tr>
<td>TMSK1</td>
<td>EQU</td>
<td>$22</td>
<td>OC1I...I4/05I, IC1I, ..., IC3I</td>
</tr>
<tr>
<td>TFLG1</td>
<td>EQU</td>
<td>$23</td>
<td>OC1F...I4/05F, IC1F, ..., IC3F</td>
</tr>
<tr>
<td>ORG</td>
<td></td>
<td>$1040</td>
<td></td>
</tr>
<tr>
<td>IC1MOD</td>
<td>RMB</td>
<td>1</td>
<td>s/w mode flag: ff-off, 0-1st, 1-last edge</td>
</tr>
<tr>
<td>IC1DUN</td>
<td>RMB</td>
<td>1</td>
<td>flag: 0—not done, 1—pulse measured</td>
</tr>
<tr>
<td>FSTEDG</td>
<td>RMB</td>
<td>2</td>
<td>time of first edge (16-bits)</td>
</tr>
<tr>
<td>PW</td>
<td>RMB</td>
<td>2</td>
<td>pulse width (16-bits)</td>
</tr>
</tbody>
</table>
ORG $2000  ; prog starts in EVB RAM @ $2000

* initialization section
LDAA #$7E  ; JMP (extented) opcode
STAA PVIC1  ; EVB's IC1 pseudo vector
LDX $SRIC1  ; address of IC1 service routine
STX PVIC1+1  ; finish JMP instruction to IC1 routine

* Main section of pulse width measurement
TOP
LDX #REGBAS  ; point to register block
LDAA #%00010000
STAA TCTL2,X  ; EDG1B:EDG1A=0:1, detect IC1 rising edge
LDAA #$FF
STAA IC1MOD  ; FF-IC1 off;
CLR IC1DUN  ; signal pulse not done
BCLR TFLG1,X $FB  ; clear old IC1F flag (if any)
BSET TMSK1,X $04  ; enable IC1 interrupts
CLI  ; enable interrupts
WAIT
LDAA IC1DUN  ; sets after pulse done
BEQ WAIT  ; loop till pulse has been timed
SEI  ; pulse done, disable interrupts
*  ; display pulse width, etc.
BRA TOP  ; go to top of main & repeat
*SRIC1-IC1 service routine

SRIC1  LDX    #REGBAS   *point to register block
        INC    IC1MOD   *$FF->0 @ 1st edge, 0->1 @ 2nd
        BNE    NO1ST    *if not 0, this is trailing edge

*process leading edge of pulse
LDD    TIC1,X   *read time of first edge
STD    FSTEDG   *save till next capture

*reconfig IC1 for falling edge
BCLR   TCTL2,X $30  *EDGLB:EDGLA->0:0
BSET   TCTL2,X $20  *EDGLB:EDGLA->1:0
BRA    OU2IC1     *done processing first edge

*process trailing edge of pulse
NO1ST  LDD    TIC1,X   *get time of trailing edge
        SUBD   FSTEDG   *time of 2nd minus time of 1st
        STD    PW      *store pulse width in hex number cycles.
BCLR   TCTL2,X $30  *disable IC1
LDAA   #1
STAA   IC1DUN     *signal pulse measured

OU2IC1 BCLR   TFLG1,X $FB  *clear IC1F
RTI
Output Compare

- Content of Output Compare Register is compared to content of Free-running counter register, when a match occurs, signal is output through that Output Compare pin
- Used for outputting waveforms to control actuators or is used to generate time delays for I/O functions
Output Compare

• An output will cause:
  - One or several port A pins to change status
  - Corresponding OCxF flags to set
  - Interrupt to occur

• Write to high-order byte of TOC register inhibits new compare for one bus cycle, to prevent erroneous comparison.
  - E.g. FF0F->00FF, no interrupt occurs @000F
**Output Compare**

Normal Pin Control Associated registers:

TOC1~TOC5, TFLG1, TMSK1, TCTL1

<table>
<thead>
<tr>
<th>TOC1~TOC5</th>
<th>b7</th>
<th>b0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>TFLG1</th>
<th>OC1F</th>
<th>OC2F</th>
<th>OC3F</th>
<th>OC4F</th>
<th>OC5F</th>
<th>IC1F</th>
<th>IC2F</th>
<th>IC3F</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>TMSK1</th>
<th>OC1I</th>
<th>OC2I</th>
<th>OC3I</th>
<th>OC4I</th>
<th>OC5I</th>
<th>IC1I</th>
<th>IC2I</th>
<th>IC3I</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>TCTL1</th>
<th>OM2</th>
<th>OL2</th>
<th>OM3</th>
<th>OL3</th>
<th>OM4</th>
<th>OL4</th>
<th>OM5</th>
<th>OL5</th>
</tr>
</thead>
</table>

Configuration:

<table>
<thead>
<tr>
<th>OMx</th>
<th>0Lx</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>OCx does not affect pin (OC1 still may)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Toggle OCx</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Drive OCx low</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Drive OCx high</td>
</tr>
</tbody>
</table>
Output Compare

- Advanced I/O Pin Control Using OC1
  - Allows one output compare to simultaneously control the states of up to five output pins
  - Can also be configured to control pin(s) that are being controlled by one of the other four OC functions (OC1 has priority when compares occur at the same time)
  - OC1M specifies which port A outputs are to be used
  - OC1D specifies what data is placed on these port pins.
  - OC1 can only affect PA7 pin if it is configured as output pin (by setting DDRA7 bit in PATCL)

<table>
<thead>
<tr>
<th>b7</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC1M</td>
<td>OC1M7</td>
</tr>
<tr>
<td>OC1D</td>
<td>OC1D7</td>
</tr>
</tbody>
</table>

Ref. PA7/PA1 | PA6/OC2 | PA5/OC3 | PA4/OC4 | PA3/OC5 | PA2/IC1 | PA1/IC2 | PA0/IC3

$100C

$100D
Output Compare

- Forced Output Compares
  - Useful to force
    - Output Compare earlier than it was scheduled
    - Actions taken as result of forced compare is same as if there were match btw OCx and TCNT, except that corresponding interrupt status flag bits are not set

- Normally is not used on an output compare function that is programmed to toggle its output on a successful compare, because normal compares immediately before or after forced compare can result in undesirable operation.

<table>
<thead>
<tr>
<th>CFORC</th>
<th>FOC1</th>
<th>FOC2</th>
<th>FOC3</th>
<th>FOC4</th>
<th>FOC5</th>
<th>b7</th>
<th>b0</th>
<th>$100B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>b7</td>
<td>b0</td>
<td>$100B</td>
</tr>
</tbody>
</table>
Pulse Accumulator

- A 8-bit counter can be read/written to at any time
- Can be configured to operate as event counter or for gated time accumulation

Figure 11-1 Pulse Accumulator Operating Modes
Pulse Accumulator

• Associated Registers

<table>
<thead>
<tr>
<th>Register</th>
<th>Bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMSK2</td>
<td>TOI</td>
<td>PAOVI PAII</td>
</tr>
<tr>
<td></td>
<td>RTII</td>
<td>PR1 PR0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$1024</td>
</tr>
<tr>
<td>TFLG2</td>
<td>TOF</td>
<td>PAOVF PAIF</td>
</tr>
<tr>
<td></td>
<td>RTIF</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$1025</td>
</tr>
<tr>
<td>PACTL</td>
<td>DDRA7</td>
<td>PAEN PAMOD</td>
</tr>
<tr>
<td></td>
<td>PEDGE</td>
<td>RTR1 RTR0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$1026</td>
</tr>
<tr>
<td></td>
<td></td>
<td>b7 b0</td>
</tr>
<tr>
<td>PACNT</td>
<td></td>
<td>$1027</td>
</tr>
</tbody>
</table>

- DDRA7: 0=input, 1=output. (normally configured as input when PA is used)
- PAEN: 0=PA disabled, 1=PA enabled
- PAMOD: 0=event counter, 1=gated time accumulation
- PEDGE
  - event counter: 0=PAI falling edge increments counter, 1= rising edge increments
  - gated time: 0= Zero volt input into PAI inhibits counting, 1= 5 volt input into PAI inhibits counting
- PAOVI, PAOVF: PA overflow interrupt enable and flag
- PAII, PAIF: PA input edge interrupt enable and flag
• **Event Counting Mode**
  - Events must be translated into rising/falling edges on PAI to be counted
  - PAMOD=0, counts active edge of PAI
  - Can cause interrupts after N events - writing N’s 2’s compliment to PACNT
  - Can count more than 256 events by tracking the number of overflows.
  - Example: Products on assembly line can be counted using light emitter/detector pair.
Pulse Accumulator

- **Gate Time Accumulation Mode**
  - PACNT increments every 64th E-clock cycle when PAI pin is active.
  - PAMOD=1, PEDGE controls inhibiting PAI pin level
  - Can be used to accumulate total time pin was active over series of pulses
  - Common use is to measure pulse width (easier than using IC)
  - Interrupt function:
    - Overflow interrupt is useful to generating signals longer than 8-bit counter range
    - PAI edge interrupt is useful for signaling end of timing period
Timers

Pulse Accumulator

• Example: Generate interrupt at specified time
  - Using gated time accumulation (PAMOD=1) to set pulse accumulator to interrupt after 5ms
    • Calculate time for one E/64 cycle
    • Divide delay by time for one E/64 cycle
    • Take 2’s complement and store in PACNT
    • When input goes to active level, counter will increment until overflow
Conclusion

• Applications of HC11 Timer systems
  - RTI:
    • Generates time delay
  - Input capture:
    • To measure pulse width of signal
  - Output compare:
    • To drive DC motor (such as in lab 5)
  - Pulse Accumulator
    • Product counter in assembly line

• Questions
References

- The IC Cookbook
  » Delton T. Horn
- A tutorial for control and timing applications
  » Commandant John R. Vig
- IC Applications Handbook
  » Arthur H. Seidman
- Handbook ok Microcircuit and applications
  » Stout and Kaufman
- The M68HC11 Microcontroller-Applications in Control, Instrumentation, and Communication
  » Michael Kheir
- M68HC11 E Series Technical Data
- M68HC11 Reference Manual