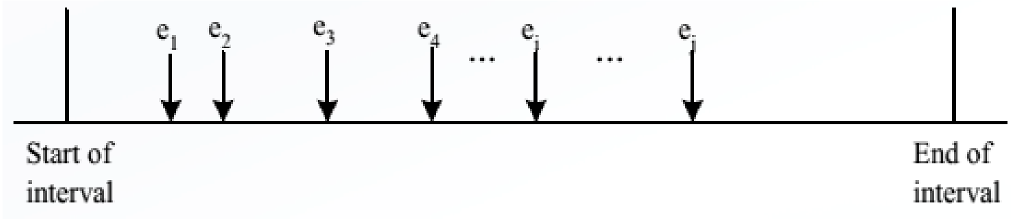
Event Counting with the HCS12

The HCS12 can be programmed to count the number of reoccurring 'events' in some application using Timer functions. An event can be considered to be a signal edge (low to high or high to low voltage transition). These events/pulses (the e_i's in the diagram) can be counted by an event counter. The HCS12 refers to an event counter as a Pulse Accumulator function.

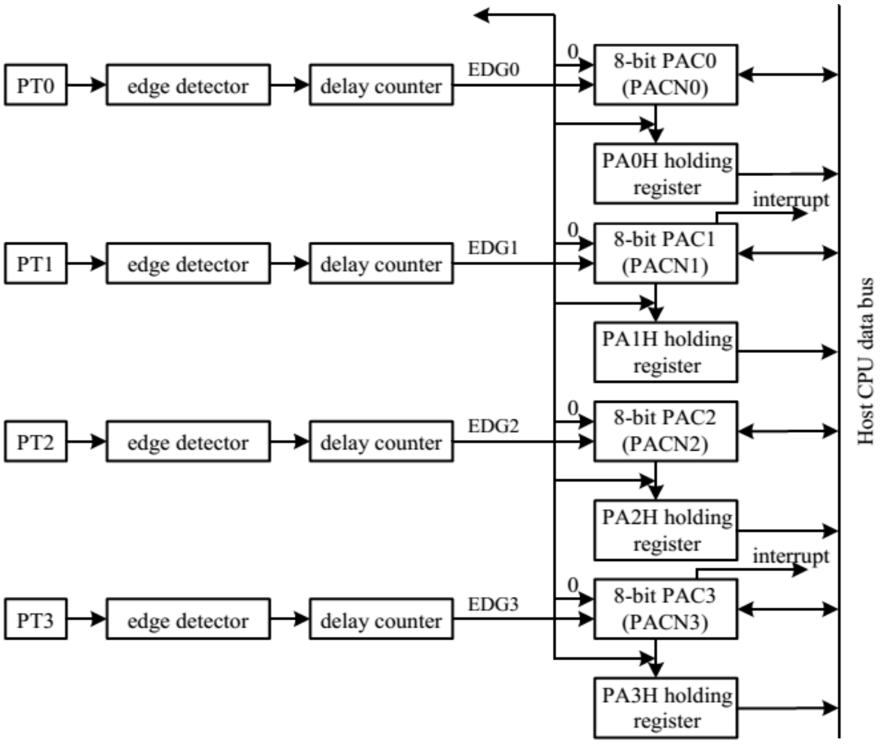


HCS12 Pulse Accumulator

The HCS12 has four 8 bit and two 16 bit Pulse Accumulators. The 8 bit Pulse Accumulator can count up to 255 pulses and the 16 bit Pulse Accumulator can count up to 65,535 pulses. Pulses can be counted on a rising or falling edge.

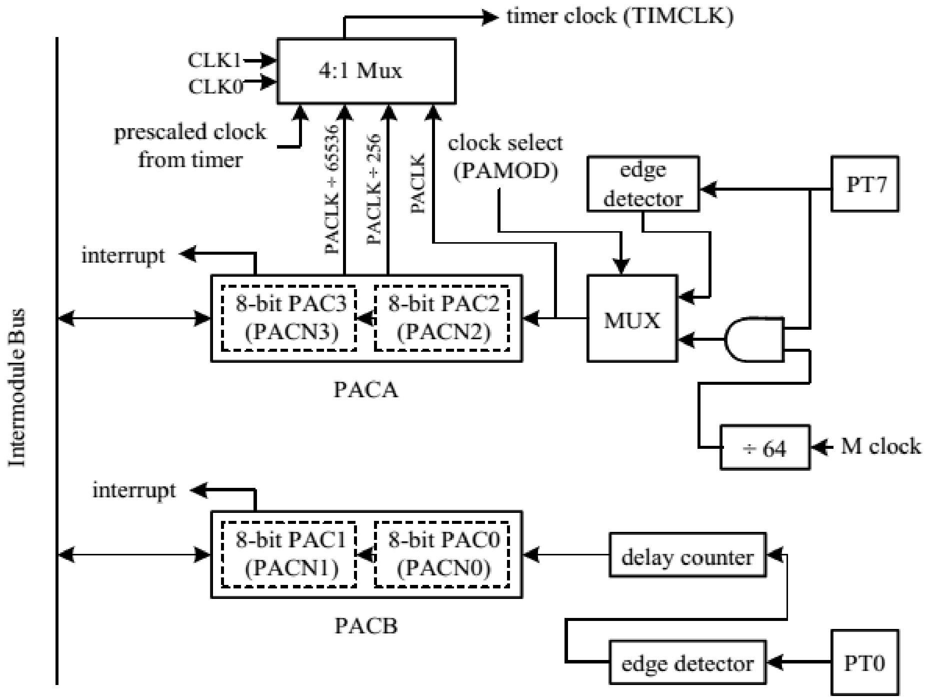
8 Bit Pulse Accumulator

The 8 bit Pulse Accumulators are referred to as Pulse Accumulators 0 to 3 and are called PACN0 to PACN3. Each of PACN0 to PACN3 has a Port T pin associated with them (PT0 – PT3).



16 Bit Pulse Accumulator

The HCS12 has two 16 bit Pulse Accumulators. These are referred to as PACA (Pulse Accumulator A) and PACB (Pulse Accumulator B). Pulse Accumulator A is formed by cascading the two 8 bit Pulse Accumulators PACN3 and PACN2 together, is controlled by PACTL (Pulse Accumulator A Control Register) and is accessed via pin PT7. Pulse Accumulator B is formed by cascading the two 8 bit Pulse Accumulators PACN1 and PACN0 together, is controlled by PBCTL (Pulse Accumulator B Control Register) and is accessed via pin PT0.



Configuration Modes

There are 4 possible configurations for these pulse accumulators

1. Two 16-bit pulse accumulators PACA and PACB
2. One 16-bit pulse accumulator PACA and two 8-bit pulse accumulators - PAC1 and PAC0
3. One 16-bit pulse accumulator PACB and two 8-bit pulse accumulators - PAC3 and PAC2
4. Four 8-bit accumulators PAC3 to PAC0

Modes of Operation

There are two modes of operation - Event counting mode and Gated-time accumulation mode. The 16-bit PACA can operate in both modes but the 16-bit PACB and all four 8-bit pulse accumulators can operate only in the Event counting mode.

Event counting mode

In this mode the 8 and 16-bit Pulse Accumulators count the number of events arriving at their designated pin.

Gated-time accumulation mode

Only the 16 bit PACA can operate in this mode. As long as the PT7 signal is active (can be high or low), the PACA counter is clocked by a free-running E ÷64 signal.

Configuring an 8 Bit Pulse Accumulator

The ICPAR (Input Control Pulse Accumulator Control Register) register is used to enable the 8 bit Pulse Accumulators.

Input Control Pulse Accumulator Control Register (ICPAR Register) - at address $0068

				PA3EN	PA2EN	PA1EN	PA0EN

PA3EN to PA0EN Pulse Accumulator Enables

0 = 8 bit Pulse Accumulator is disabled 1 = 8 bit Pulse Accumulator is enabled

Sample Programs

Assembly Code

```
;*****************************************
;*  This program counts 1 Hz pulses fed to pin PTO
;*  which is connected to PACN0 (8 bit Pulse Accumulator)
;*  The state of the count is displayed on PORT B
;*******************************************

; Register addresses
TCTL4:    equ   $4B     ;address of TCTL4 register
PORTB:    equ   $01     ;address of PORTB register
DDRB:     equ   $02     ;address of DDRB for PORT B register
DDRT:     equ   $242    ;address of DDRT for PTH register
ICPAR:    equ   $68     ;address of ICPAR register
PACN0:    equ   $65     ;address of PACN0 register

      org $2000
    movb  #$FF,DDRB   ;configure PORT B as output
    bclr  DDRT,$00    ;make PT0 input
    movb  #$01,ICPAR  ;enable PACN0 (pin PT0) as pulse accumulator
    movb  #$01,TCTL4  ;capture the rising edge of PT0 signal
    ldaa  #0          ;initialize count in A to 0
    staa  PACN0       ;initialize Pulse Acc with value of count
again:
    ldaa PACN0        ;read PACN0 on every count
    staa PORTB        ;send count to Port B
    cmpa #255         ;test for maximum count
    bne  again        ;when count reaches 0 exit loop

; dress up the code by printing out a message
    swi
    end
```

Additional Program Comments

The programmer will need to determine the pin number for PT0 on the Demo board of the HCS12 mini-board. This can be found from appropriate documentation.

An alternative program code fragment is shown. Assume that you want the code to count 200 pulses and then exit the loop. Load the value of 200 (C8 in hex) into A, find the 2’s complement of A (38 in hex) and use this value. The Pulse Accumulator counts up from 0011 1000 until it reaches 1111 1111 (199 counts). On the next count (200) the value becomes 0000 0000 which sets the zero flag and the program exits the loop.

```
    movb  #$FF,DDRB   ;configure PORT B as output
    bclr  DDRT,$00    ;make PT0 input
    movb  #$01,ICPAR  ;enable PT0 as pulse acc
    movb  #$01,TCTL4  ;capture the rising edge of PT0 signal
    ldaa  #200        ;initialize count in A to 200
    nega              ;find 2’s comp of count
    staa  PACN0       ;initialize Pulse Acc with 2's comp of count
again:
    ldaa PACN0        ;read PACN0 on every count
    staa PORTB        ;send count to Port B
    bne  again        ;when count reaches 0 exit loop

    swi
    end
```

C Code

```
//   This program reads the QUAD encoder output
//   of a DC motor using the 8 bit Pulse Accumulator
//   feature of the HCS12 at PT3 pin 10.
//   The count is displayed on Port H
//   The C program counts 225 counts and prints out a message
//   when the count is complete.

#include <hidef.h>           /* common defines and macros */
#include "derivative.h"      /* derivative-specific definitions */
#include <stdio.h>

void main(void) {

   unsigned int count=225;     //set count value
   DDRH = 0xFF;                //init Port H as O/P
   DDRT &= ~0x08;              //init PT3 as input
   ICPAR = 0x08;               //enable PT3 as pulse accum
   TCTL4 = 0x40;               //set to capture rising edge
   count=~count + 1;           //2's comp of count value
   PACN3=count;                //load count into Pulse Acc reg

 while(PACN3)
    {
       PTH = PACN3;            //o/p value until count=0
    }
   printf("The count is complete \n\r");   /* \n=LF  \r=CR  */

   asm("swi");
}
```

Configuring a 16 Bit Pulse Accumulator

16 Bit Pulse Accumulator A - PACA

The PACTL (Pulse Accumulator A Control Register) register is used to control the 16 bit Pulse Accumulator A. The 16 bit Pulse Accumulator Count register associated with PACA is referred to as PACN32 and is at address $0062.

Pulse Accumulator A Control Register (PACTL Register) - at address $0060

0	PAEN	PAMOD	PEDGE	CLK1	CLK0	PAOVI	PAI
bit 7							bit 0

PAEN Pulse Accumulator A enable

0 = Disable PACA 1 = Enable PACA

PAMOD PACA is used in event counter or gated timer mode

0 = use in event timer mode
1 = use in gated timer mode

PEDGE PACA counts up on falling or rising edge

0 = increment on falling edge
1 = increment on rising edge

CLK1, CLK0 use the value 0 0

PAOVI PACA Overflow Interrupt enable

0 = interrupt inhibited
1 = interrupt requested if PAOVF is set

PAI PACA Input Interrupt enable

0 = interrupt inhibited
1 = interrupt requested if PAIF is set

Pulse Accumulator A Flag Register (PAFLG Register) - at address $0061

						PAOVF	PAIF

PAOVF — Pulse Accumulator A Overflow Flag

Set when the 16-bit pulse accumulator A overflows from $FFFF to $0000,or when 8-bit pulse accumulator 3 (PAC3) overflows from $FF to $00.

This bit is cleared automatically by a write to the PAFLG register with bit 1 set.

PAIF — Pulse Accumulator Input edge Flag

Set when the selected edge is detected at the PT7 input pin. In event mode the event edge triggers PAIF and in gated time accumulation mode the trailing edge of the gate signal at the PT7 input pin triggers PAIF.

This bit is cleared by a write to the PAFLG register with bit 0 set. Any access to the PACN3, PACN2 registers will clear all the flags in this register when TFFCA bit in register TSCR1 is set.

16 Bit Pulse Accumulator B - PACB

The PBCTL (Pulse Accumulator B Control Register) register is used to control the 16 bit Pulse Accumulator B. The 16 bit Pulse Accumulator Count register associated with PACB is referred to as PACN10 and is at address $0064.

Pulse Accumulator B Control Register (PBCTL Register) - at address $0070

	PBEN					PBOVI	

bit 7

bit 0

Pulse Accumulator B Flag Register (PBFLG Register) - at address $0071

						PBOVF	

PBOVF — Pulse Accumulator B Overflow Flag

Set when the 16-bit pulse accumulator B overflows from $FFFF to $0000, or when 8-bit pulse accumulator 3 (PAC1) overflows from $FF to $00.

This bit is cleared automatically by a write to the PAFLG register with bit 1 set.

Sample Code

Assembler Code

```
;*********************************************
;*     This program reads pulses connected at
;*     PT7 (pin 6) using the PACA 16 bit Pulse Accumulator.
;*
;*  The program detects a pulse at PT7 and then
;*  increments PACN32 the 16 bit Pulse Accumulator register.
;*  The value of PACN32 is sent to PORT H
;*  until the count reaches zero
;**********************************************

;register addresses

PTH:      equ   $260       ;address of Port H register
DDRH:     equ   $262       ;address of Port H control register
DDRT:     equ   $242       ;address of Port T control register
PACTL:    equ   $60        ;address of PACA control register
PACN32:   equ   $62        ;address of PACA counter register

;parameter values

Count:    equ   240        ;count value

       org $2000

    movb  #$FF,DDRH     ;configure PORT H as output
    bclr  DDRT,$80      ;make PT7 on Port H input
    movb  #$50,PACTL    ;enable 16 bit event counter, NoInt, rising edge
    ldd   #Count        ;load count into Reg D
    coma                ;find 1's comp of upper half of D
    comb                ;find 1's comp of lower half of D
    addd  #1            ;find 2's comp of D
    std   PACN32        ;initialize PACA counter register with 2's comp of count
again:
    ldd  PACN32         ;read PACN32
    stab PTH            ;send lower half to LEDs
    bne again
    swi
    end
```

C Code

```
/********************************************
;*     This program reads pulses connected at
;*     PT7 (pin 6) using the PACA 16 bit Pulse Accumulator.
;*
;*  The program detects a pulse at PT7 and then
;*  increments PACN32 the 16 bit Pulse Accumulator register.
;*  The value of PACN32 is sent to PORT H
;*  until the count reaches zero
;*********************************************/


#include <hidef.h>           /* common defines and macros */
#include "derivative.h"      /* derivative-specific definitions */
#include <stdio.h>
 void main(void) {

    unsigned int count=900;     //set count value
    count=~count+1;             //2's comp of count
    DDRH=0xFF;                  //initialize PORT H as O/P
    DDRT&=~0x80;                //make PT7 input
    PACTL=0x50;                 //enable 16 bit event count, noInt,rising edge
    PACN32=count;               //load count into Pulse Accum Count register

    while(PACN32)
    {
      PTH=PACN32;               //display value until count=0
    }

   printf("The count has been completed \n\r");   /* \n=LF  \r=CR  */

   asm("swi");
}
```

Application Note
Written by David Lloyd
Computer Engineering Program
Humber College