Real Time Clocks & Timers

Programming with Real-time clocks

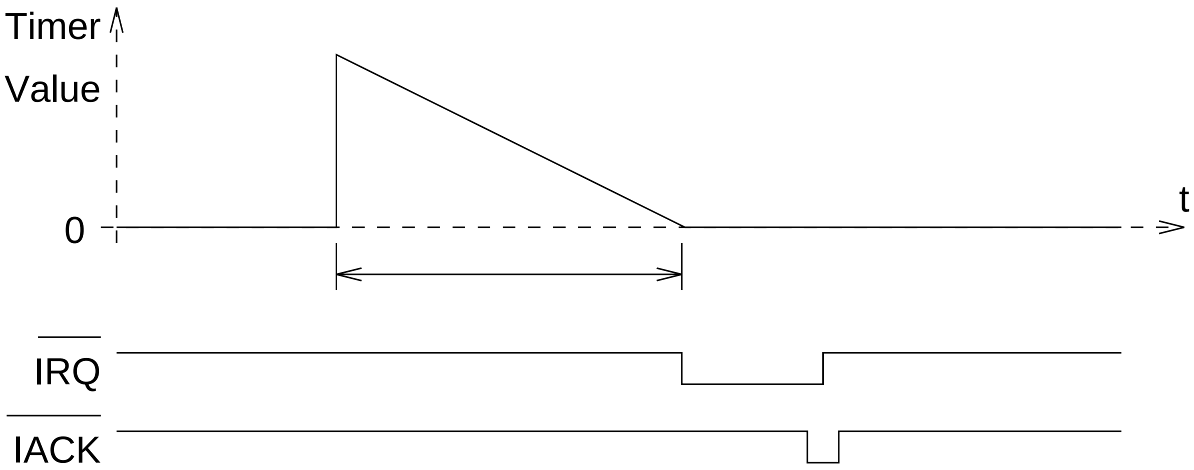
- Real-time clock is just another source of interrupts.
- Should have high priority in real-time systems
- Timing jitter must be accommodated or tolerated

Software related use of Timers

- Single event scheduled for future time
  - e.g. timeout for communications
- Regular scheduling of events
  - e.g. time-slice multi-tasking

Timer Devices

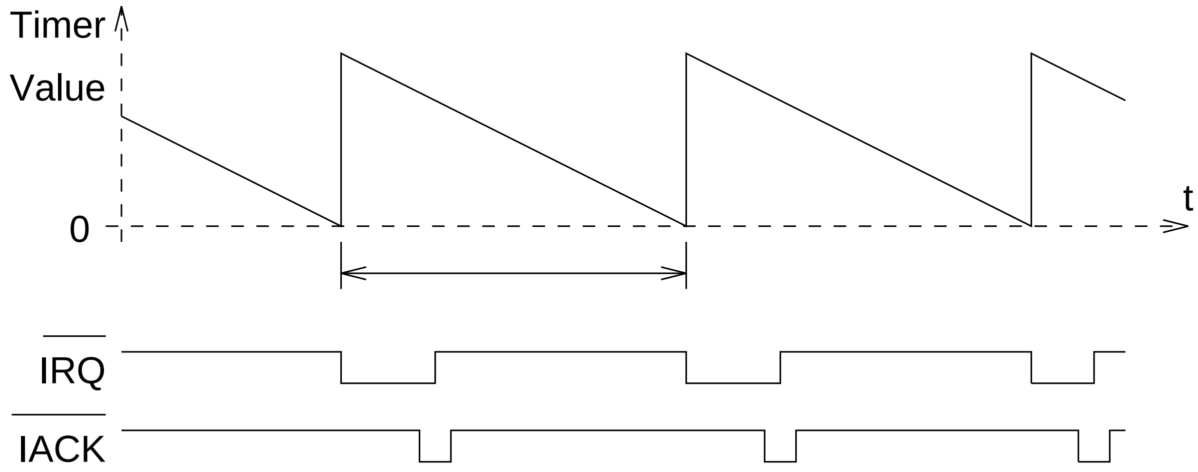
- Single shot operation
  - Provides single interrupt after specified timeout



  - counter is loaded with the desired value and then set running.
  - an interrupt occurs when the counter reaches zero.

Timer Devices

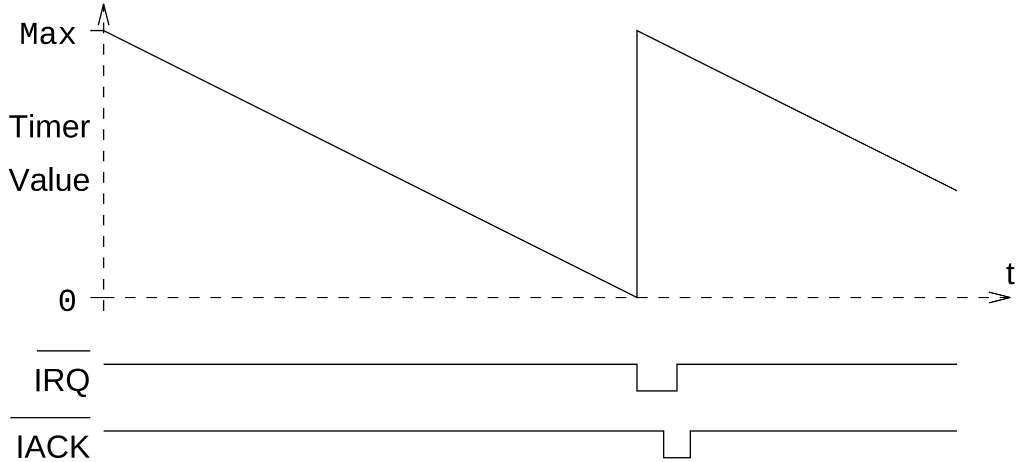
- Continuous operation
  - Provides regular interrupts at specified intervals



  - each time zero is reached, the counter is re-loaded with the interval value and countdown continues.

Timer Devices

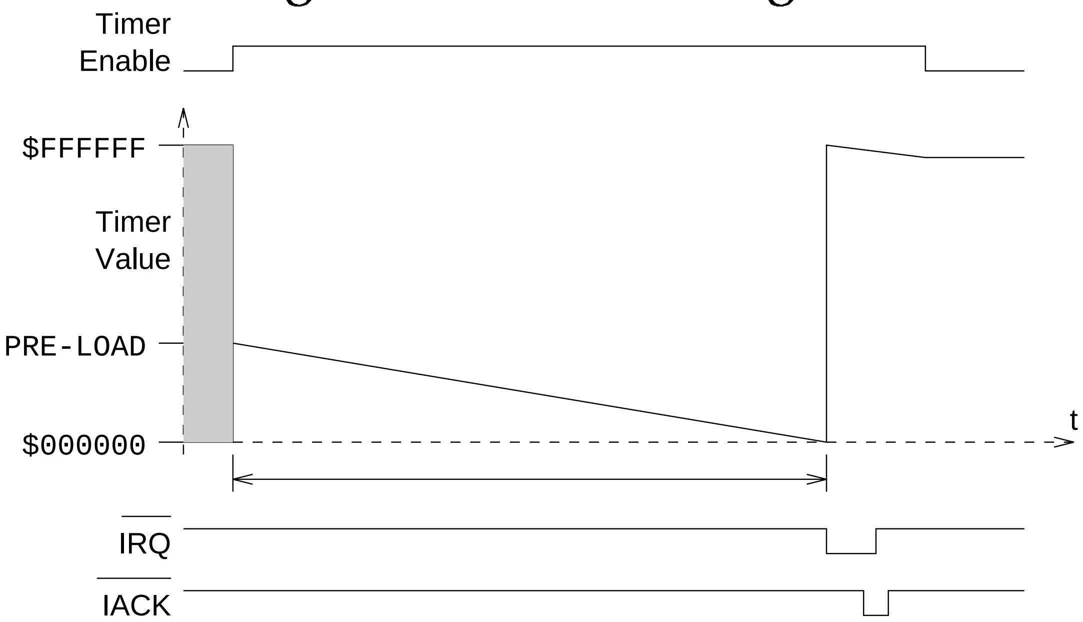
- System Clock
  - May be consulted at any time, giving an absolute value for system time.



  - pre-load value is set to maximum.
  - interrupt routine is responsible for keeping count of number of full sequences.

68230 Peripheral Interface / Timer

- 24 bit counter
- provides extra timing information in single shot mode

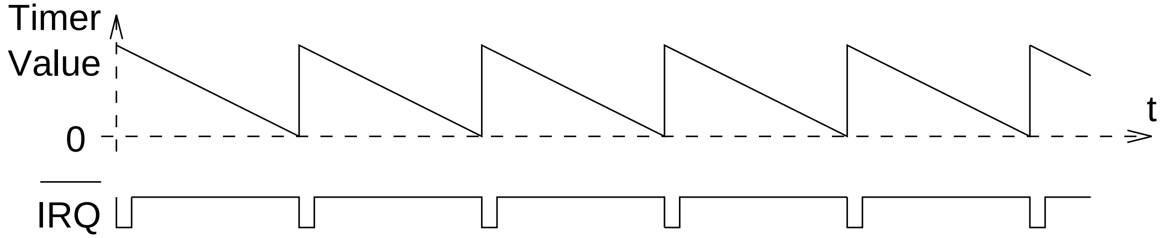


- in fact this is really another continuous mode with a very long second cycle.

Sharing a Countdown Timer

While running as a system clock the timer may be accessed by many routines. Timer sharing for single-shot or periodic interrupts is somewhat harder.

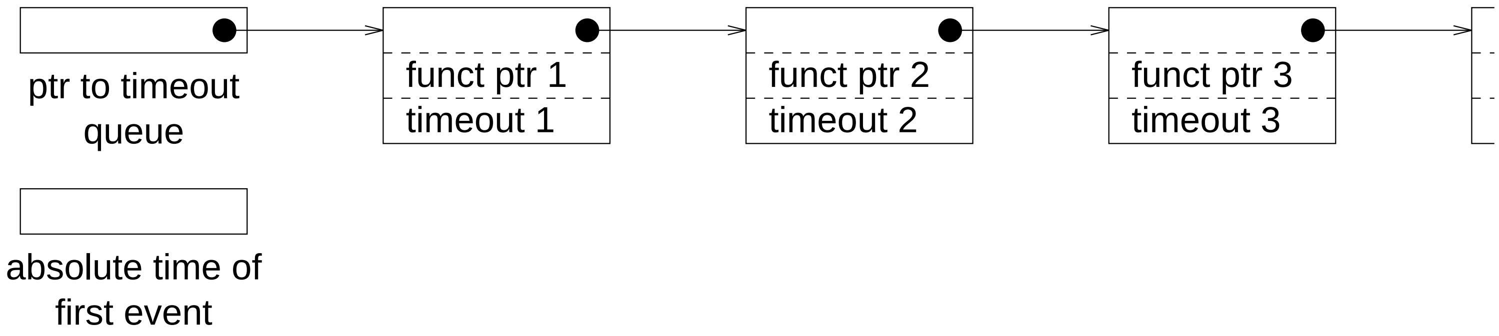
- Tick based timer queues



  - the timer triggers an interrupt at regular intervals
  - pending timer events are stored in a queue
  - on each tick the interrupt routine checks to see if another timer event is due
  - tick based queues offer limited resolution

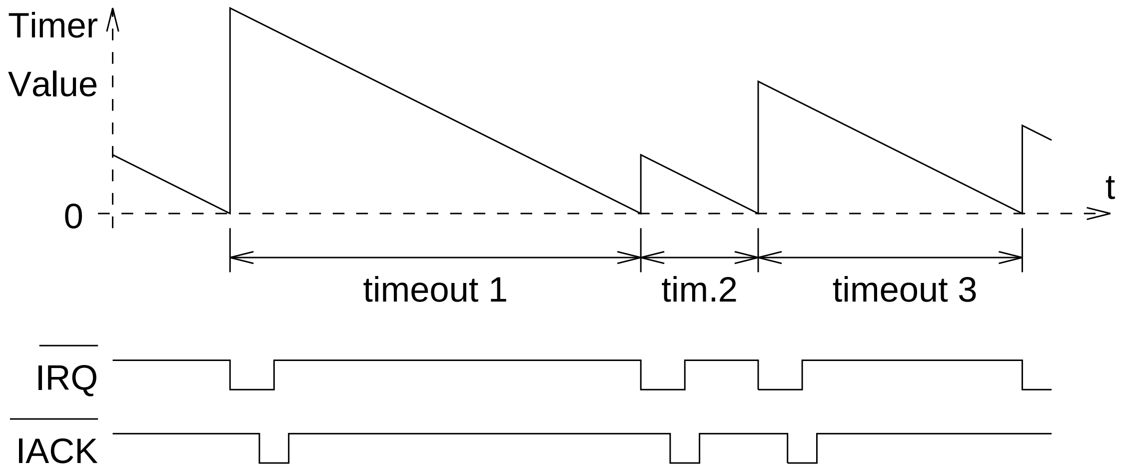
Timer Queues

For accurate timing we must use a difference based timer queue:



- each queue entry contains a pointer to an event service routine
- timeout values are the time differences between adjacent events
- we need only know the absolute time for the first event

Timer Queues



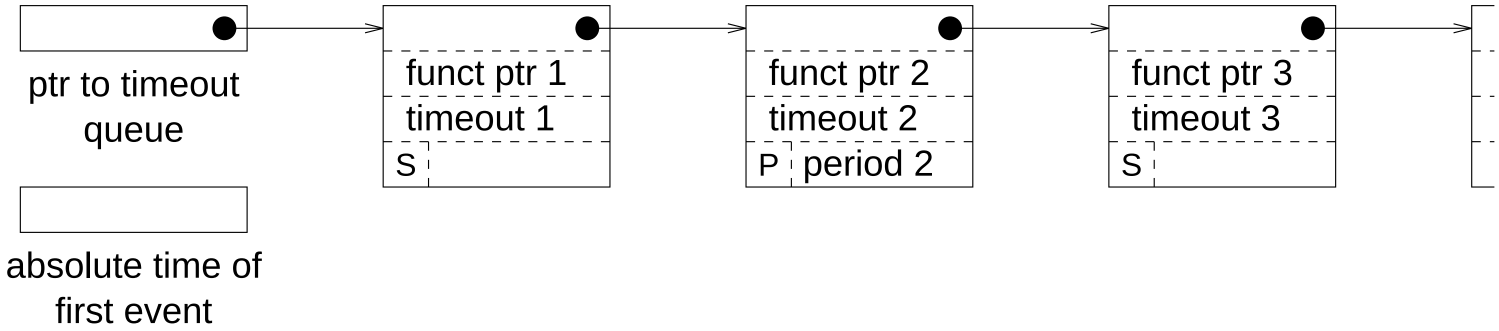
- each time the counter reaches zero
  - timeout $n + 1$ is automatically loaded into the counter
  - interrupt routine loads timeout $n+2$ into the pre-load registers
  - interrupt routine adjusts absolute time register
  - interrupt routine starts function n

Queue manipulation

- Adding an event
  - find place in queue
  - calculate difference timeout
  - re-calculate difference timeout for following event
- Adding an event behind the first queue element
  - here we must modify the pre-load registers during counting
- Adding an event at the front of the queue
  - we must re-start the timer with a new counter value

Queue manipulation

- Periodic events
  - we need an extra field to indicate periodic events



  - periodic events must be added back into the queue when serviced

Hardware & Timers

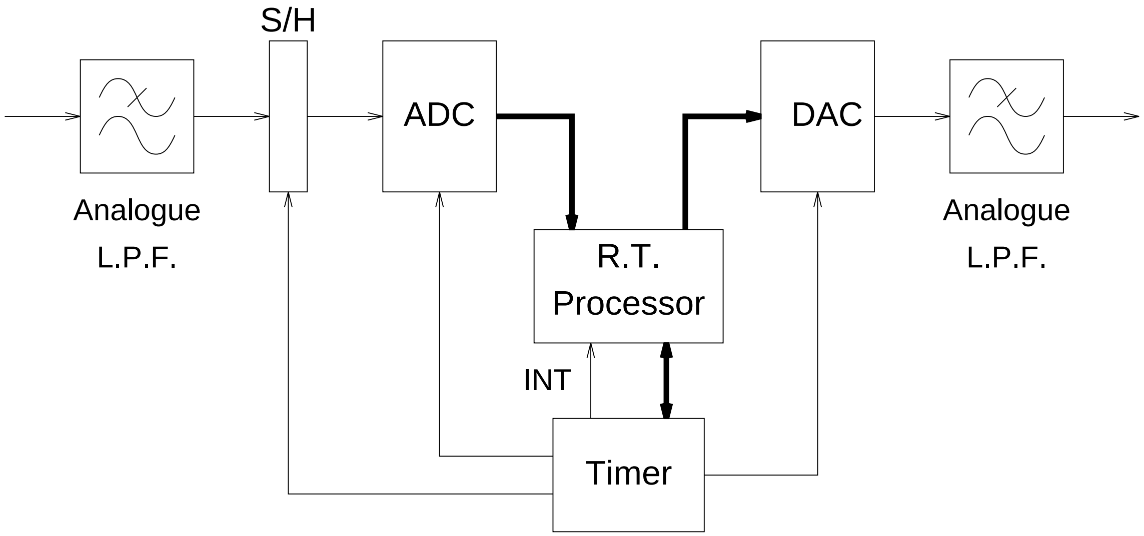
Timing accuracy is limited by unpredictable and variable response times – *timing jitter*.

For many hardware related problems, a greater accuracy is required. We can achieve this by connecting the hardware directly to the timer.

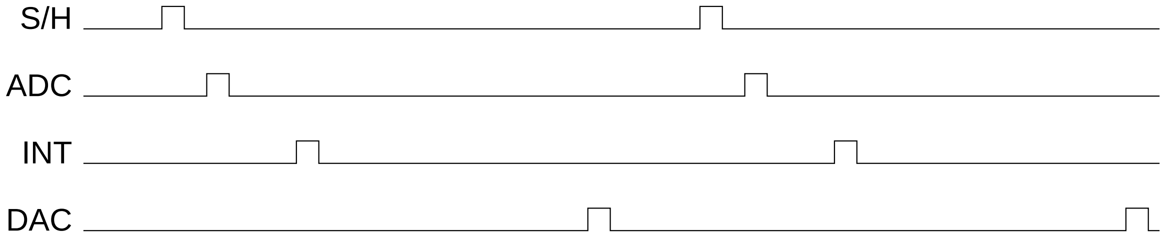
- Timer input from hardware
  - start/stop timer
  - external clock input — counts pulses
  - enable timer
- Timer output to hardware
  - event strobes
  - square wave generation

Hardware-Attached Timing

- Digital filter illustrating hardware attached timing

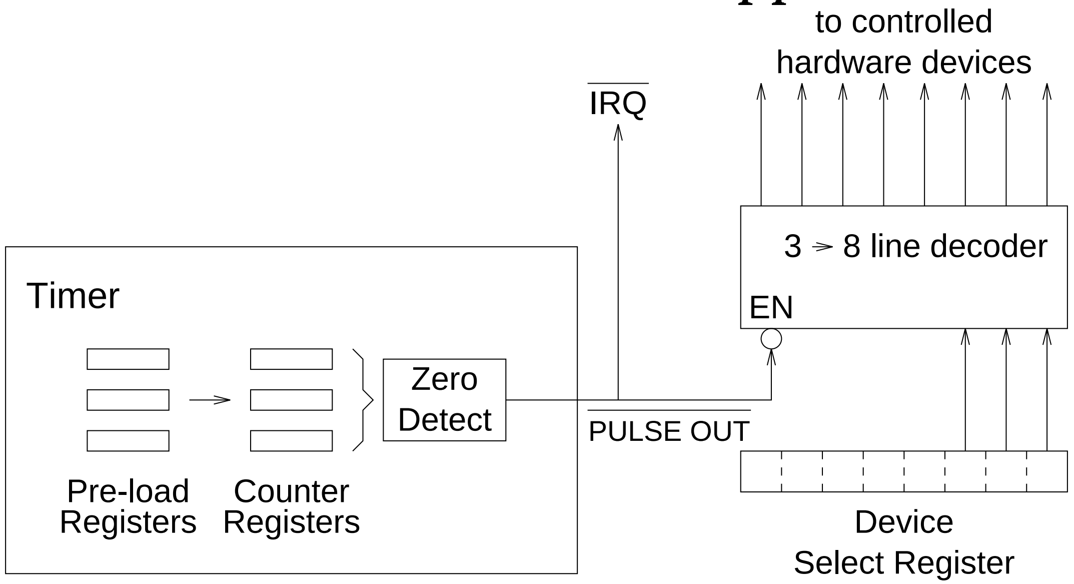


- Timer provides separate regular strobe signals for different devices



Hardware-Attached Timing

Sharing a timer for "hardware-attached" applications:



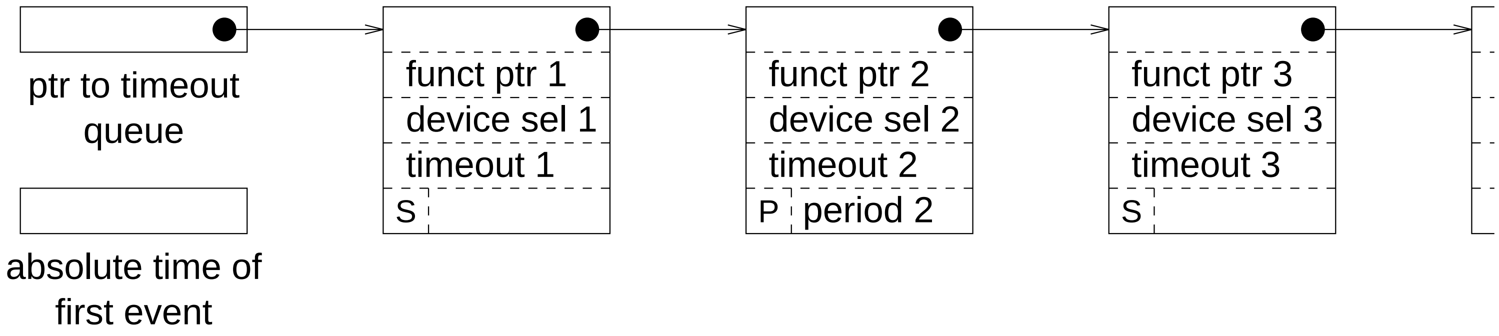
The 68230 PI/T provides a single configurable timer input line and a single timer output line[1].

- a device select register indicates which device receives next strobe
- the decoder activates the required strobe line when enabled

[1] this doubles as the interrupt request

Hardware-Attached Timers

- the device select field must be added to our timer queue information



- the device select register is updated by the master service routine after the interrupt is cleared