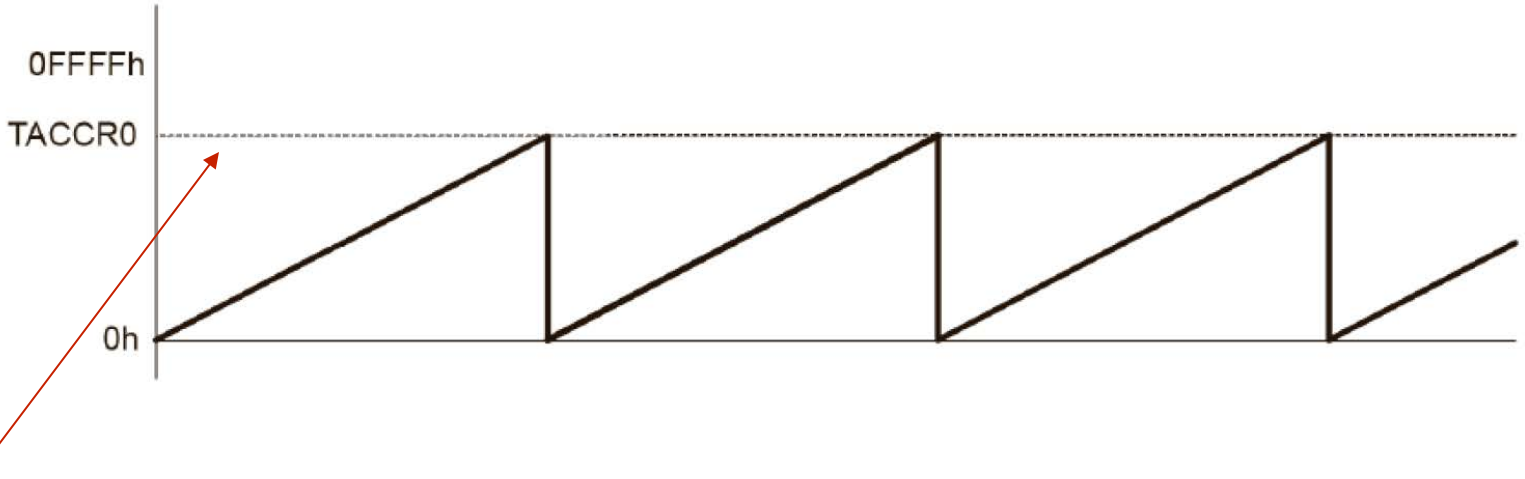
Timers

- MSP430F2272 has two 16-bit timers
  - So can count to 0xFFFF
- Independent of the CPU
- Three capture/compare registers
  - TACCR0, TACCR1 & TACCR2 can generate interrupts when timer meets certain conditions (many modes/highly configurable)

Register	Short Form	Register Type	Address	Initial State
Timer_A control	TACTL	Read/write	0160h	Reset with POR
Timer_A counter	TAR	Read/write	0170h	Reset with POR
Timer_A capture/compare control 0	TACCTL0	Read/write	0162h	Reset with POR
Timer_A capture/compare 0	TACCR0	Read/write	0172h	Reset with POR
Timer_A capture/compare control 1	TACCTL1	Read/write	0164h	Reset with POR
Timer_A capture/compare 1	TACCR1	Read/write	0174h	Reset with POR
Timer_A capture/compare control 2	TACCTL2†	Read/write	0166h	Reset with POR
Timer_A capture/compare 2	TACCR2†	Read/write	0176h	Reset with POR
Timer_A interrupt vector	TAIV	Read only	012Eh	Reset with POR

Timers

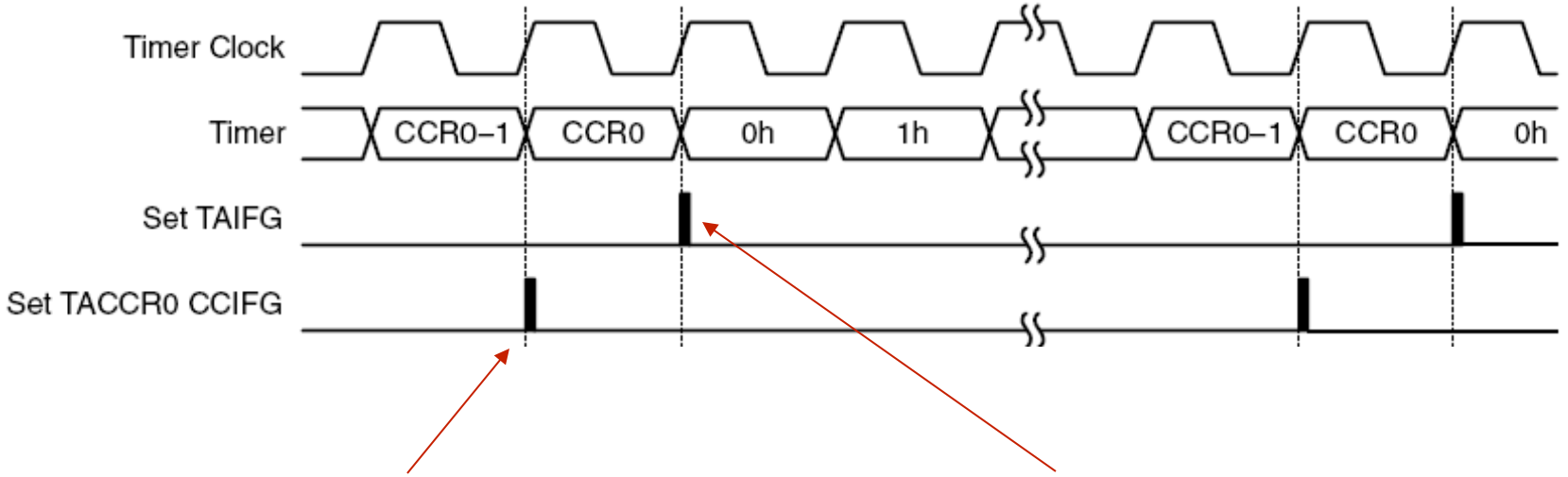
- 16-bit counter register TAR (and TBR)
  - Increments/decrements with each rising edge of the clock signal
- Four operating modes
  - Stop (not very interesting)
  - Up



timer counts UP from zero to TACCR0 then resets to 0 and starts over

Timers & Interrupts

- Four operating modes
  - Stop (not very interesting)
  - Up



The TACCR0 interrupt flag CCIFG is set when the timer count reaches the value in TACCR0.

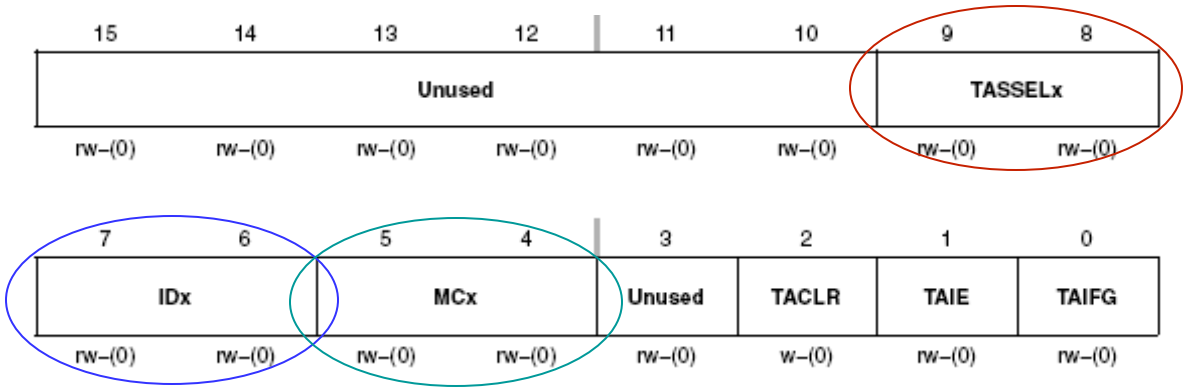
The Timer_A interrupt flag TAIFG is set when the timer count overflows the value in TACCR0 and gets back to 0.

Timer Configuration

Register	Short Form	Register Type	Address	Initial State
Timer_A control	TACTL	Read/write	0160h	Reset with POR
Timer_A counter	TAR	Read/write	0170h	Reset with POR
Timer_A capture/compare control 0	TACCTL0	Read/write	0162h	Reset with POR
Timer_A capture/compare 0	TACCR0	Read/write	0172h	Reset with POR
Timer_A capture/compare control 1	TACCTL1	Read/write	0164h	Reset with POR
Timer_A capture/compare 1	TACCR1	Read/write	0174h	Reset with POR
Timer_A capture/compare control 2	TACCTL2†	Read/write	0166h	Reset with POR
Timer_A capture/compare 2	TACCR2†	Read/write	0176h	Reset with POR
Timer_A interrupt vector	TAIV	Read only	012Eh	Reset with POR

- TACTL—Timer_A Control Register
  - Used to configure how the timer runs
- TAR—Timer_A Register
  - Holds the current count of the timer
- TACCTLx—Timer_A Capture/Compare Control Register
  - Used to configure the three capture/compare modules
- TACCRx—Timer_A Capture/Compare Register
  - Holds the capture/compare values for the three capture/compare modules
- TAIV—Timer_A Interrupt Vector Register
  - Used to decode non-TACCR0 interrupts

Timers—TACTL Timer_A Control Register



Unused	Bits 15-10	Unused
TASSELx	Bits 9-8	Timer_A clock source select 00 TACLK 01 ACLK 10 SMCLK 11 INCLK
IDx	Bits 7-6	Input divider. These bits select the divider for the input clock. 00 /1 01 /2 10 /4 11 /8
MCx	Bits 5-4	Mode control. Setting MCx = 00h when Timer_A is not in use conserves power. 00 Stop mode: the timer is halted. 01 Up mode: the timer counts up to TACCR0. 10 Continuous mode: the timer counts up to 0FFFFh. 11 Up/down mode: the timer counts up to TACCR0 then down to 0000h.

We must tell Timer_A what clock signal to use. For example, if we want the default Subsystem Clock (SMCLK), set bits 9-8 = %10 = TASSEL_2 (defined in header file).

We can slow down the counter.

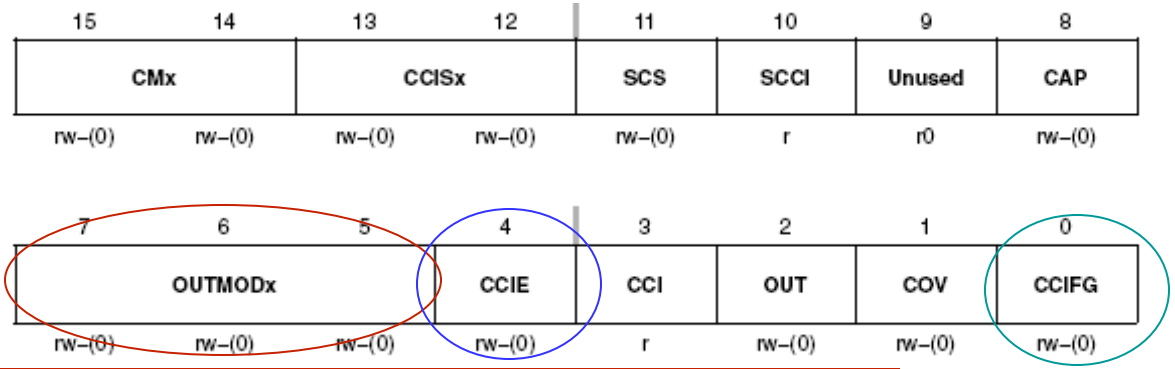
Be sure to specify counting mode!

12.3.1 TACTL, Timer_A Control Register

15	14	13	12	11	10	9	8
Unused						TASSELx	
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
IDx		MCx		Unused	TACLR	TAIE	TAIFG
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Unused	Bits 15-10	Unused
TASSELx	Bits 9-8	Timer_A clock source select 00 TACLK 01 ACLK 10 SMCLK 11 INCLK (INCLK is device-specific and is often assigned to the inverted TBCLK) (see the device-specific data sheet)
IDx	Bits 7-6	Input divider. These bits select the divider for the input clock. 00 /1 01 /2 10 /4 11 /8
MCx	Bits 5-4	Mode control. Setting MCx = 00h when Timer_A is not in use conserves power. 00 Stop mode: the timer is halted. 01 Up mode: the timer counts up to TACCR0. 10 Continuous mode: the timer counts up to 0FFFFh. 11 Up/down mode: the timer counts up to TACCR0 then down to 0000h.
Unused	Bit 3	Unused
TACLR	Bit 2	Timer_A clear. Setting this bit resets TAR, the clock divider, and the count direction. The TACLR bit is automatically reset and is always read as zero.
TAIE	Bit 1	Timer_A interrupt enable. This bit enables the TAIFG interrupt request. 0 Interrupt disabled 1 Interrupt enabled
TAIFG	Bit 0	Timer_A interrupt flag 0 No interrupt pending 1 Interrupt pending

Timers—TACCTLx Cap/Comp Control Register



OUTMODx Bits 7-5 Output mode. Modes 2, 3, 6, and 7 are not useful for TACCR0 because EQUx = EQU0.
- 000 OUT bit value
- 001 Set
- 010 Toggle/reset
- 011 Set/reset
- 100 Toggle
- 101 Reset
- 110 Toggle/set
- 111 Reset/set

These bits let you tailor the output characteristics.

CCIE Bit 4 Capture/compare interrupt enable. This bit enables the interrupt request of the corresponding CCIFG flag.
- 0 Interrupt disabled
- 1 Interrupt enabled

This bit enables the interrupt.

CCIFG Bit 0 Capture/compare interrupt flag
- 0 No interrupt pending
- 1 Interrupt pending

This bit tells you if the interrupt has occurred.

12.3.4 TACCTLx, Capture/Compare Control Register

15	14	13	12	11	10	9	8
CMx		**CCISx**		**SCS**	**SCCI**	**Unused**	**CAP**
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r	r0	rw-(0)
7	6	5	4	3	2	1	0
OUTMODx			**CCIE**	**CCI**	**OUT**	**COV**	**CCIFG**
rw-(0)	rw-(0)	rw-(0)	rw-(0)	r	rw-(0)	rw-(0)	rw-(0)

CMx Bit 15-14 Capture mode
- 00 No capture
- 01 Capture on rising edge
- 10 Capture on falling edge
- 11 Capture on both rising and falling edges

CCISx Bit 13-12 Capture/compare input select. These bits select the TACCRx input signal. See the device-specific data sheet for specific signal connections.
- 00 CCIxA
- 01 CCIxB
- 10 GND
- 11 V_{CC}

SCS Bit 11 Synchronize capture source. This bit is used to synchronize the capture input signal with the timer clock.
- 0 Asynchronous capture
- 1 Synchronous capture

SCCI Bit 10 Synchronized capture/compare input. The selected CCI input signal is latched with the EQUx signal and can be read via this bit

Unused Bit 9 Unused. Read only. Always read as 0.

CAP Bit 8 Capture mode
- 0 Compare mode
- 1 Capture mode

OUTMODx Bits 7-5 Output mode. Modes 2, 3, 6, and 7 are not useful for TACCR0, because EQUx = EQU0.
- 000 OUT bit value
- 001 Set
- 010 Toggle/reset
- 011 Set/reset
- 100 Toggle
- 101 Reset
- 110 Toggle/set
- 111 Reset/set

CCIE Bit 4 Capture/compare interrupt enable. This bit enables the interrupt request of the corresponding CCIFG flag.
- 0 Interrupt disabled
- 1 Interrupt enabled

CCI Bit 3 Capture/compare input. The selected input signal can be read by this bit.

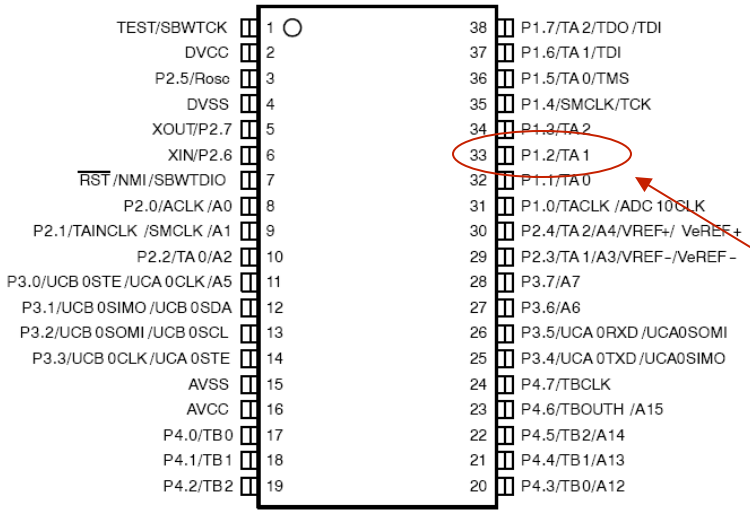
OUT Bit 2 Output. For output mode 0, this bit directly controls the state of the output.
- 0 Output low
- 1 Output high

COV Bit 1 Capture overflow. This bit indicates a capture overflow occurred. COV must be reset with software.
- 0 No capture overflow occurred
- 1 Capture overflow occurred

CCIFG Bit 0 Capture/compare interrupt flag
- 0 No interrupt pending
- 1 Interrupt pending

Timers—TACCTLx Cap/Comp Control Register

- Two ways to go
  - Timer + GPIO
    - Set/reset a GPIO pin inside timer ISR
    - Controlled by software, so requires extra CPU cycles
  - Timer output
    - Use the timer output directly
    - Can be totally controlled by hardware alone

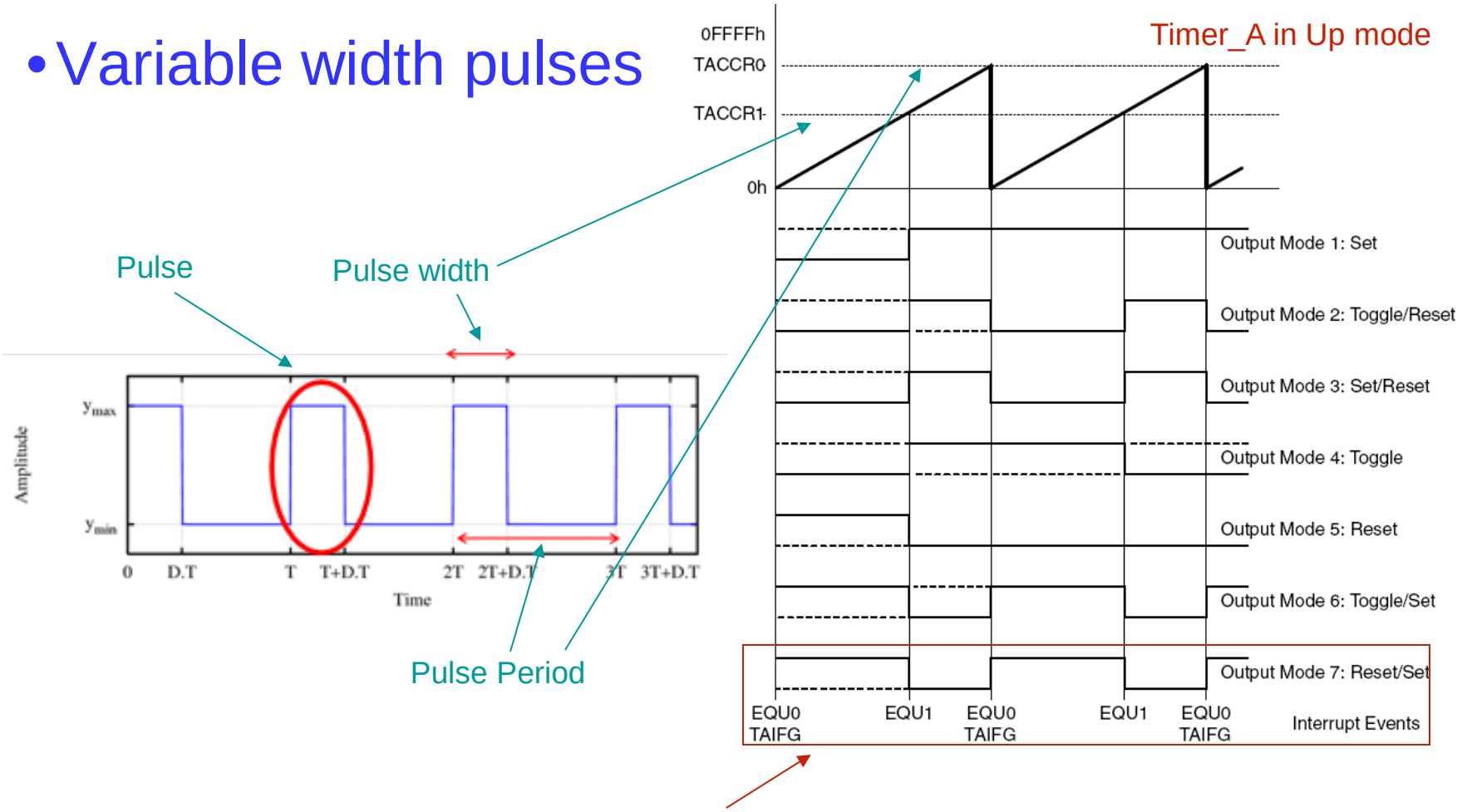


This is what you want to do for the lab!

For example: You could use Timer_A in Up mode with output mode 7 and use the timer output signal on pin TA1 to control an LED.

Timers—TACCTLx Cap/Comp Control Register

- Variable width pulses



Use TACCR0 and TACCR1 control PWM period and duty cycle.

Fourier Representation of Signals

- Any periodic waveform can be represented as an (infinite) sum of sine and cosine waveforms.
- Used to determine the frequency content (spectrum) of signals

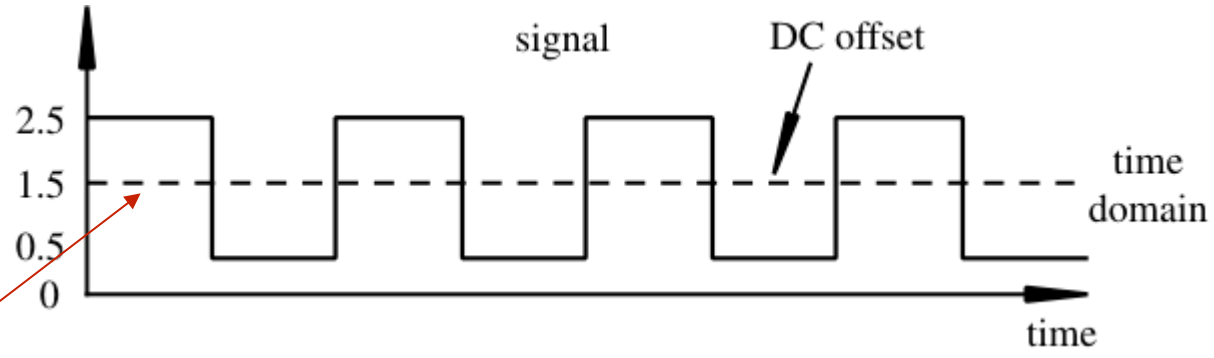
$$y(t) = A_0 + \sum_{n=1}^{\infty} \left(A_n \cos\left(n\omega t\right) + B_n \sin\left(n\omega t\right) \right)$$

$$A_n = \frac{1}{T}\int_0^T y(t)\cos\left(nwt\right)dt \qquad B_n = \frac{1}{T}\int_0^T y(t)\sin\left(nwt\right)dt$$

T = 2π/f is the period of the signal; *f* is the 1st harmonic, 2*f* is the 2nd harmonic, etc.

Frequency Domain Representation

- Square Wave



DC offset

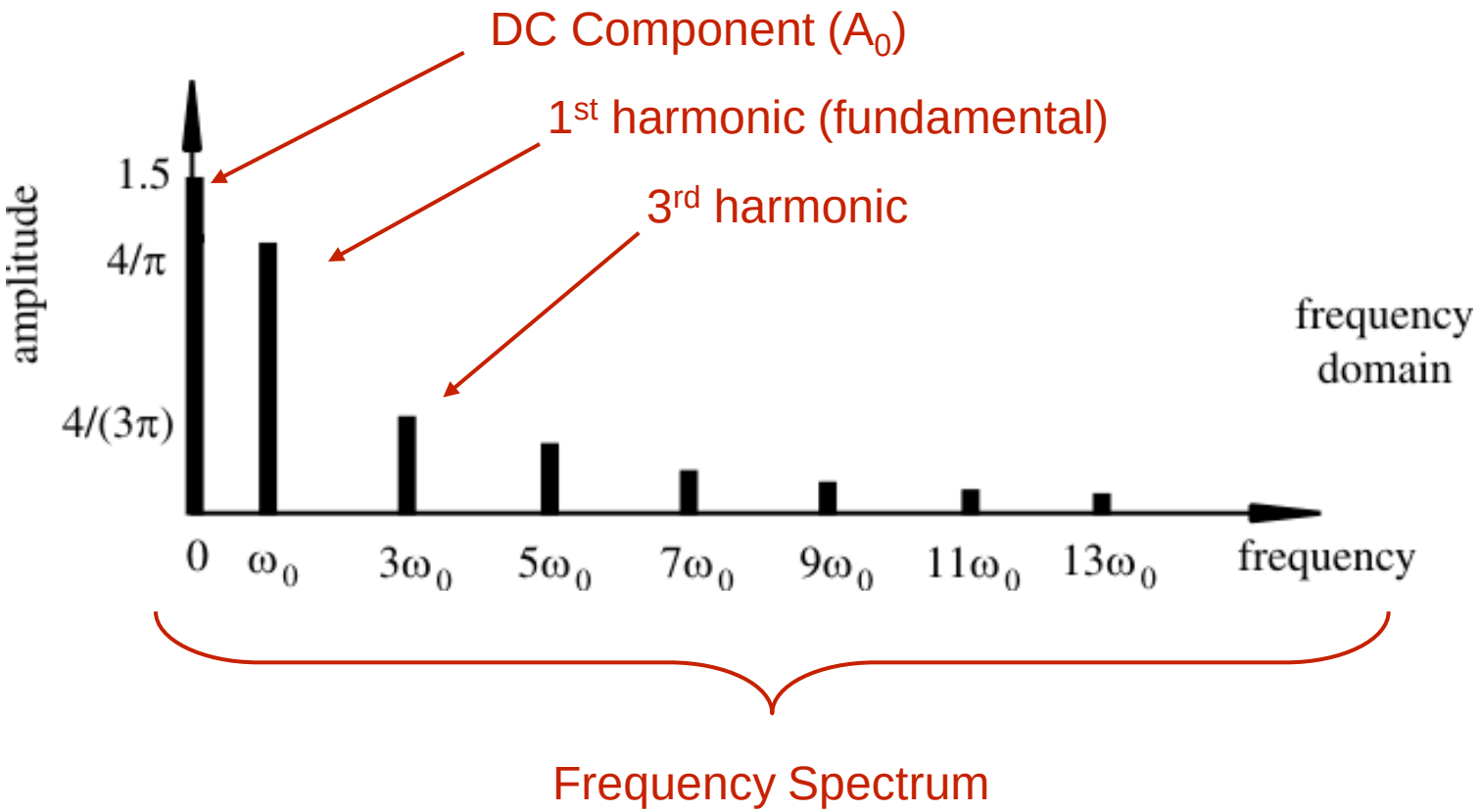
'average value'

$$y(t) = A_0 + \sum_{n=1}^{\infty} \left(A_n \cos(n\omega t) + B_n \sin(n\omega t) \right)$$

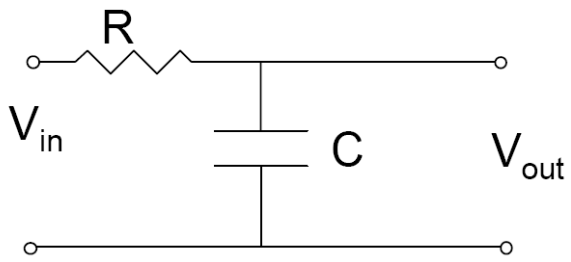
$$A_0 = \frac{1}{T} \int_0^T y(t)dt$$

Frequency Domain Representation

- Square Wave



RC Low Pass Filter



$$f_c = \frac{1}{2\pi RC}$$

$$\frac{V_o}{V_i} = \frac{1}{\sqrt{1 + (f/f_c)^2}}$$

$$\phi = -\tan^{-1}(f/f_c)$$