

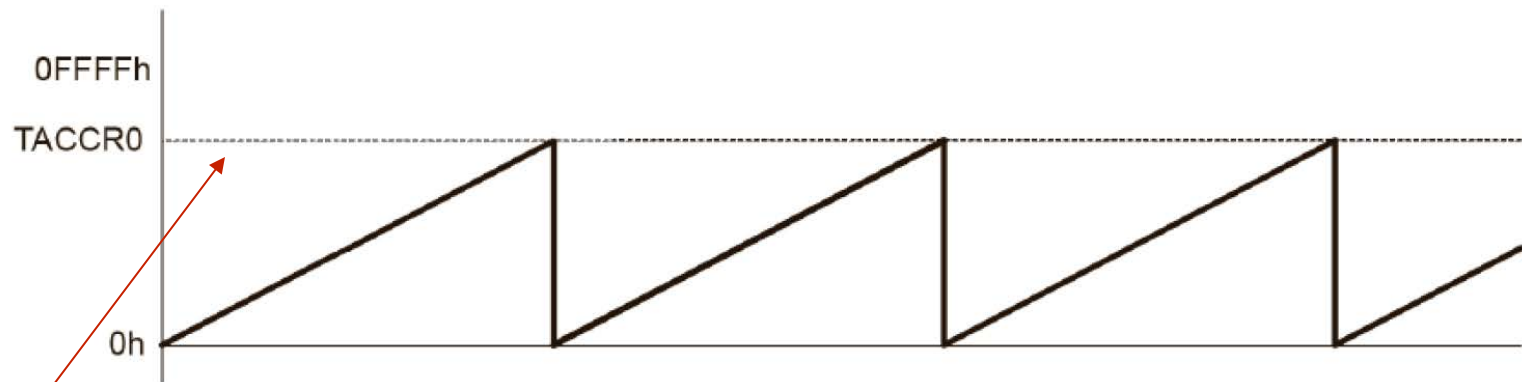
Timers

- 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

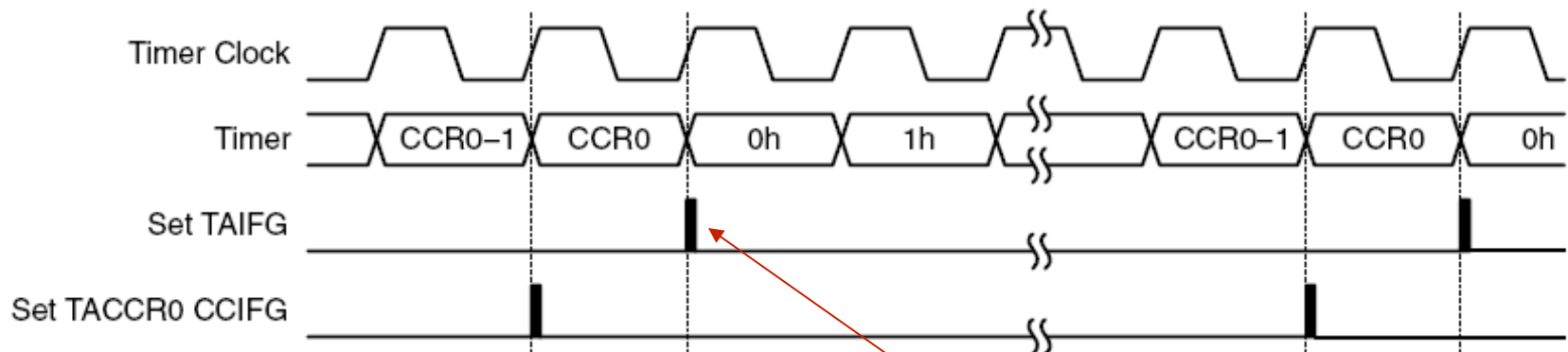


timer counts UP from zero to TACCR0 then resets to 0 and starts over

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.

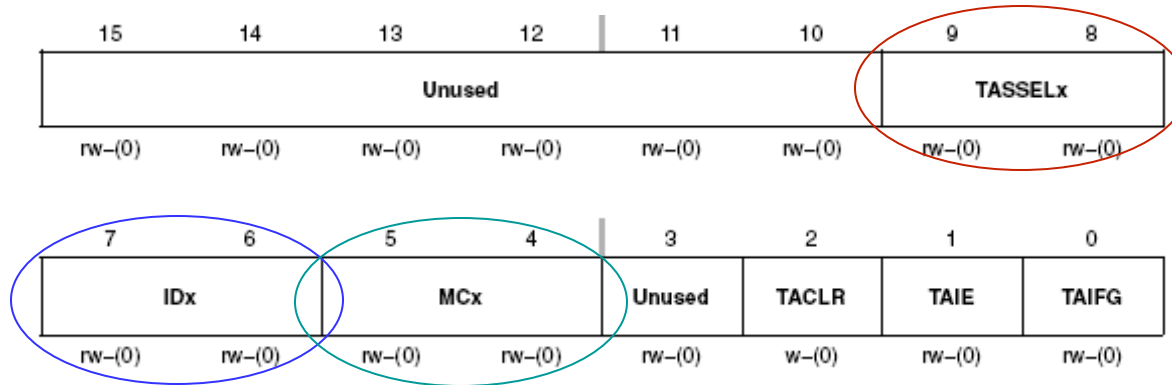
The Timer_A interrupt flag **TAIFG** is set when the timer count overflows the value in TACCR0 and gets back to 0.

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	Unused
	15-10	
TASSELx	Bits 9-8	Timer_A clock source select
		00 TACLK
		01 ACLK
		10 SMCLK
		11 INCLK

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).

IDx	Bits	Input divider. These bits select the divider for the input clock.
	7-6	00 /1
		01 /2
		10 /4
		11 /8

We can slow down the counter.

MCx	Bits	Mode control. Setting MCx = 00h when Timer_A is not in use conserves power.
	5-4	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.

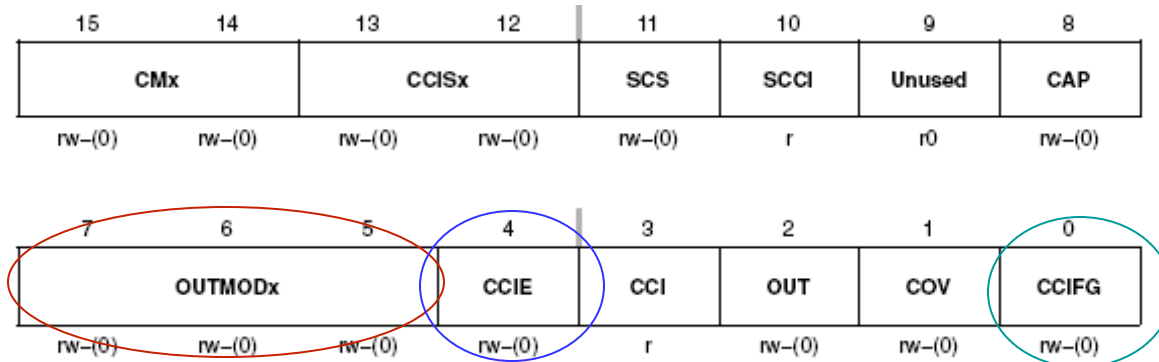
Be sure to specify counting mode!

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

CCIFG Bit 0 Capture/compare interrupt flag

0	No interrupt pending
1	Interrupt pending

This bit enables the interrupt.

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

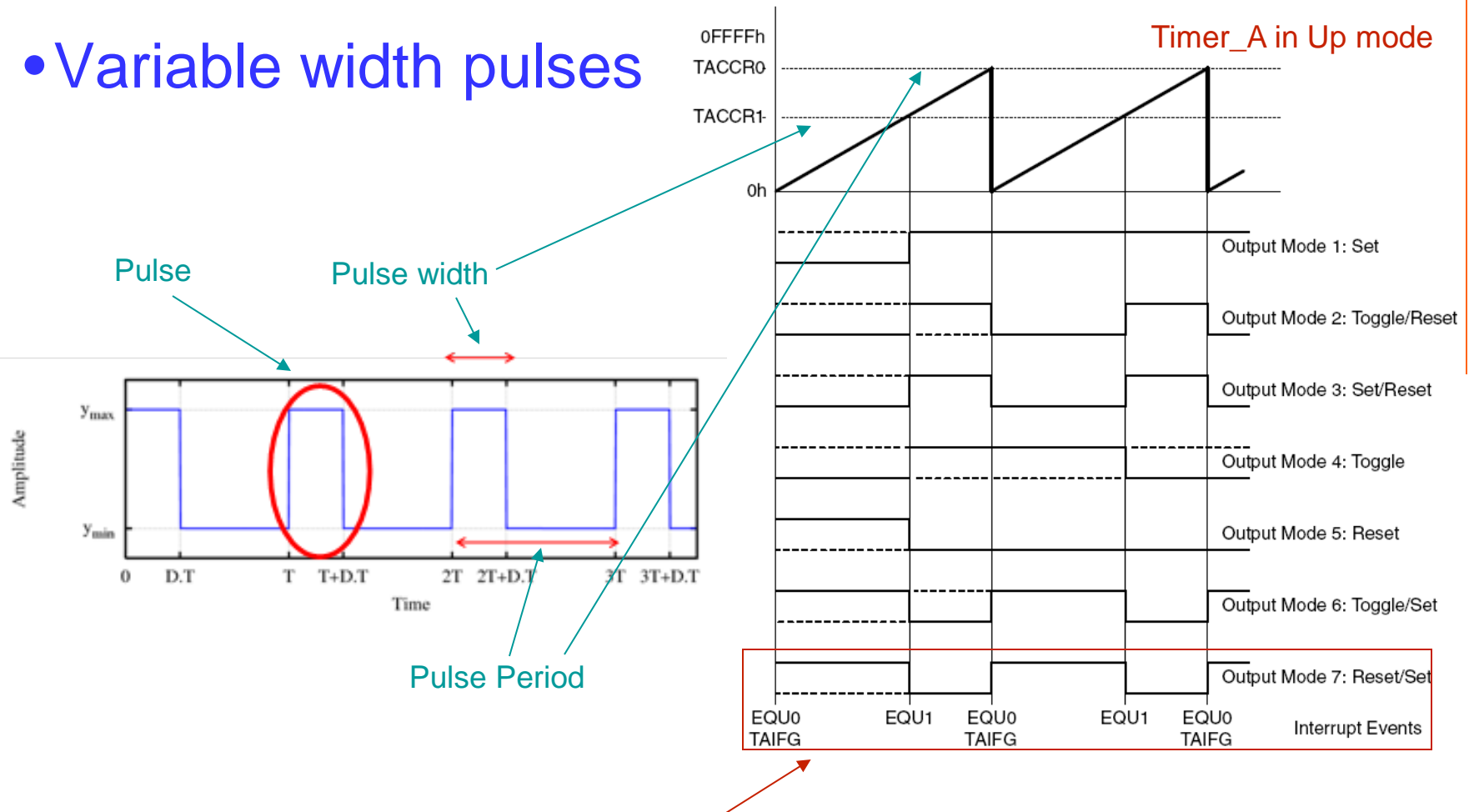
TEST/SBWTK	1	○	38	P1.7/TA2/TDO/TDI
DVCC	2		37	P1.6/TA1/TDI
P2.5/Rosc	3		36	P1.5/TA0/TMS
DVSS	4		35	P1.4/SMCLK/TCK
XOUT/P2.7	5		34	P1.3/TA2
XIN/P2.6	6		33	P1.2/TA1
RST/NMI/SBWTDIO	7		32	P1.1/TA0
P2.0/ACLK/A0	8		31	P1.0/TACLK /ADC10CLK
P2.1/TAINCLK /SMCLK /A1	9		30	P2.4/TA2/A4/VREF+ /VeREF+
P2.2/TA0/A2	10		29	P2.3/TA1/A3/VREF- /VeREF-
P3.0/UCB0STE/UCA0CLK/A5	11		28	P3.7/A7
P3.1/UCB0SIMO /UCB0SDA	12		27	P3.6/A6
P3.2/UCB0SOMI /UCB0SCL	13		26	P3.5/UCA0RXD /UCA0SOMI
P3.3/UCB0CLK/UCA0STE	14		25	P3.4/UCA0TXD /UCA0SIMO
AVSS	15		24	P4.7/TBCLK
AVCC	16		23	P4.6/TBOUTH /A15
P4.0/TB0	17		22	P4.5/TB2/A14
P4.1/TB1	18		21	P4.4/TB1/A13
P4.2/TB2	19		20	P4.3/TB0/A12

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

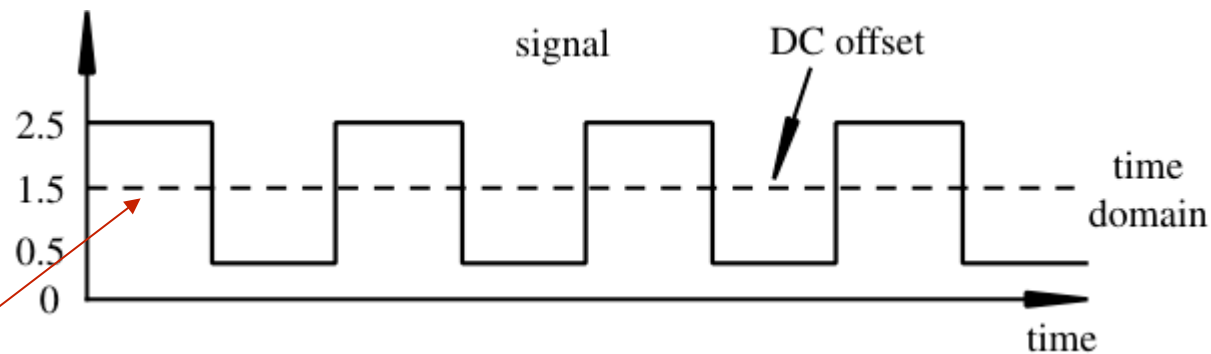
$$y(t) = A_0 + \sum_{n=1}^{\infty} (A_n \cos(n\omega t) + B_n \sin(n\omega t))$$

$$A_n = \frac{1}{T} \int_0^T y(t) \cos(n\omega t) dt \quad B_n = \frac{1}{T} \int_0^T y(t) \sin(n\omega t) dt$$

$T = 2\pi/f$ is the period of the signal; f is the 1st harmonic, $2f$ is the 2nd harmonic, etc.

Frequency Domain Representation

- Square Wave



DC offset

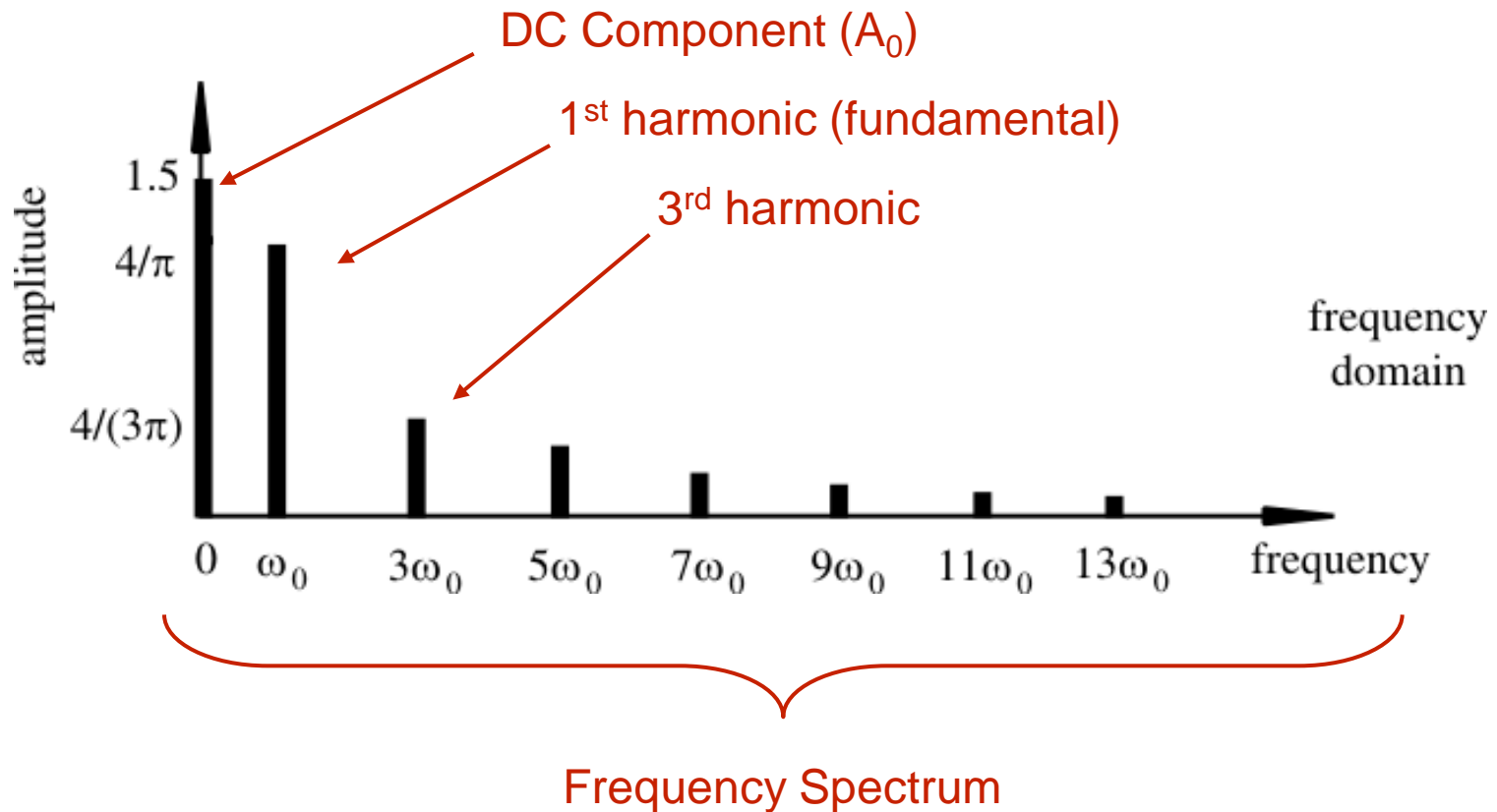
'average value'

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

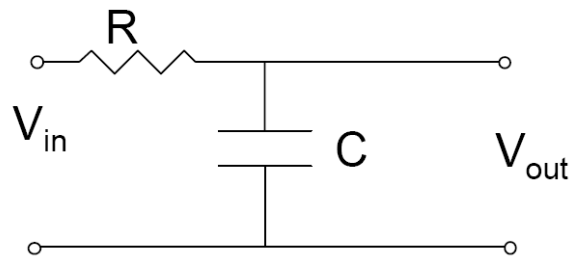
$$A_0 = \frac{1}{T} \int_0^T y(t) dt$$

Frequency Domain Representation

- Square Wave



RC Low Pass Filter



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

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

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

