Lesson 3
Timer

Lecturer: Harvard Tseng

Reset and clock control (RCC)

- APB peripheral clock enable register 1 (RCC_APB1ENR)

RCC_APB1ENR

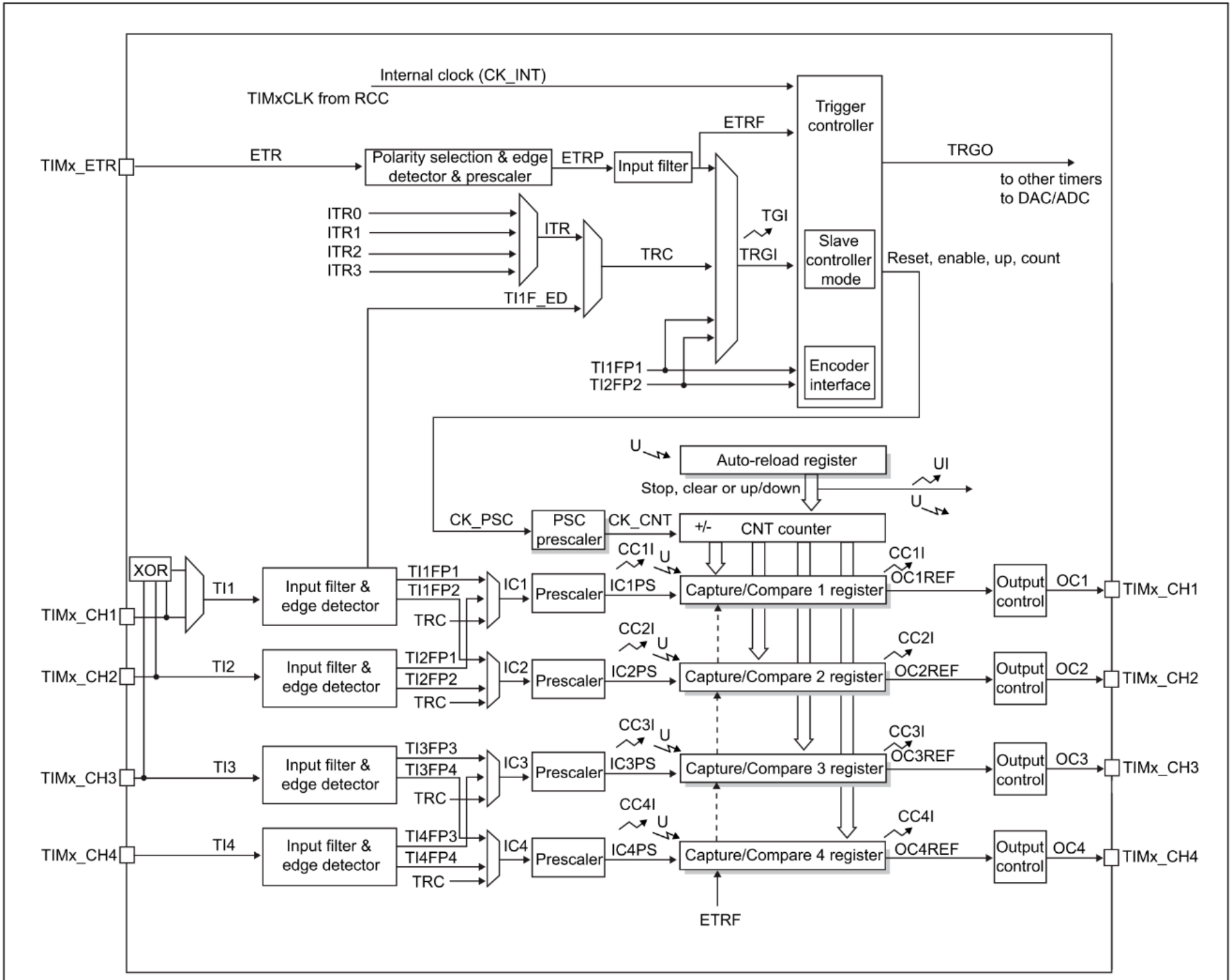
- Set and cleared by software.
  - 0: clock disable.
  - 1: clock enable.
- Writing RCC_APB1ENR_TIM2EN to enable Timer2 clock.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	CEC EN	DAC EN	PWR EN	CRS EN	Res.	CAN EN	Res.	USB EN	I2C2 EN	I2C1 EN	USART5 EN	USART4 EN	USART3 EN	USART2 EN	Res.
	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw	rw	rw	

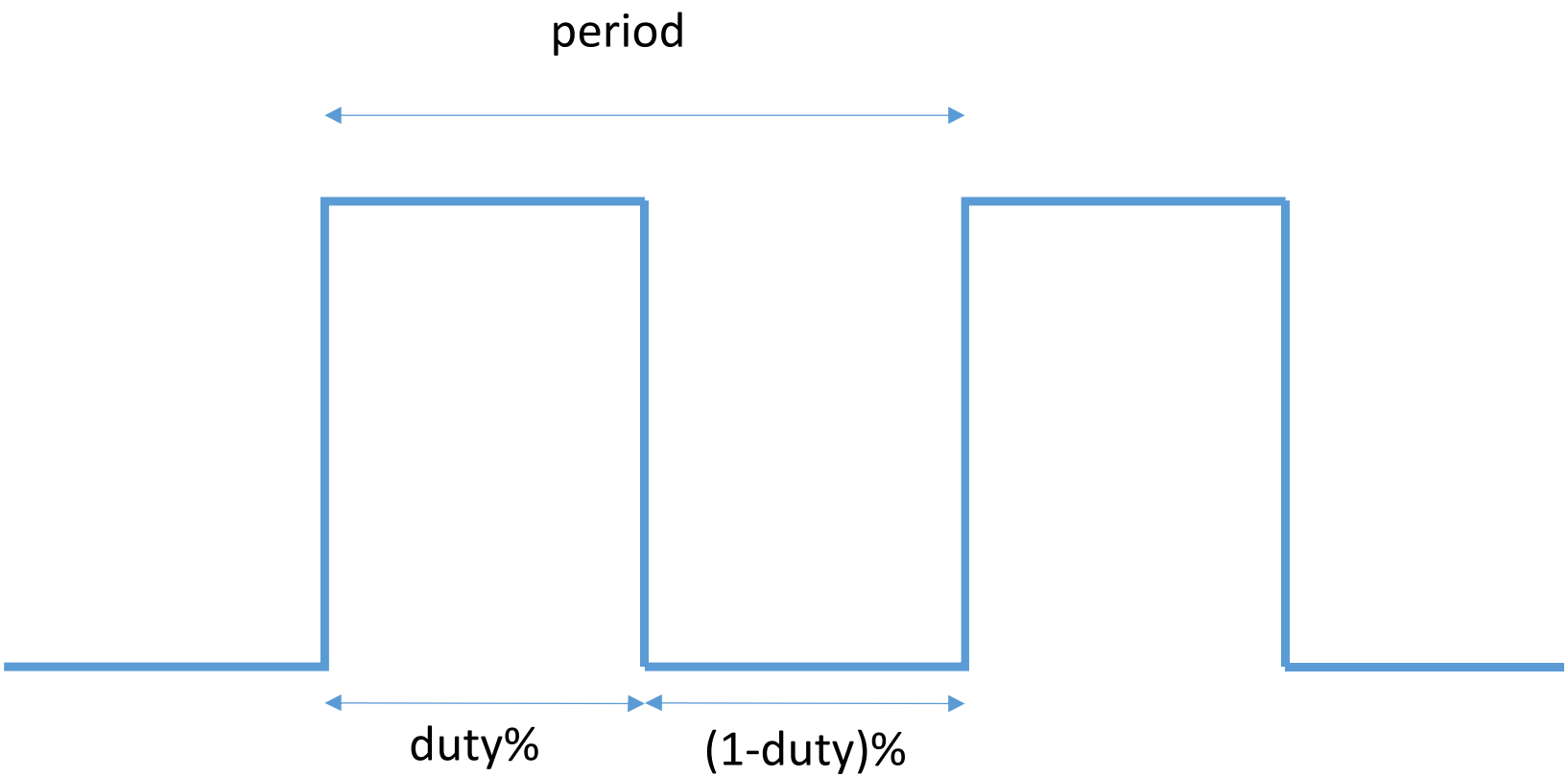
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SPI2 EN	Res.	Res.	WWDG EN	Res.	Res.	TIM14 EN	Res.	Res.	TIM7 EN	TIM6 EN	Res.	Res.	TIM3 EN	TIM2 EN
	rw			rw			rw			rw	rw			rw	rw

General-purpose timers

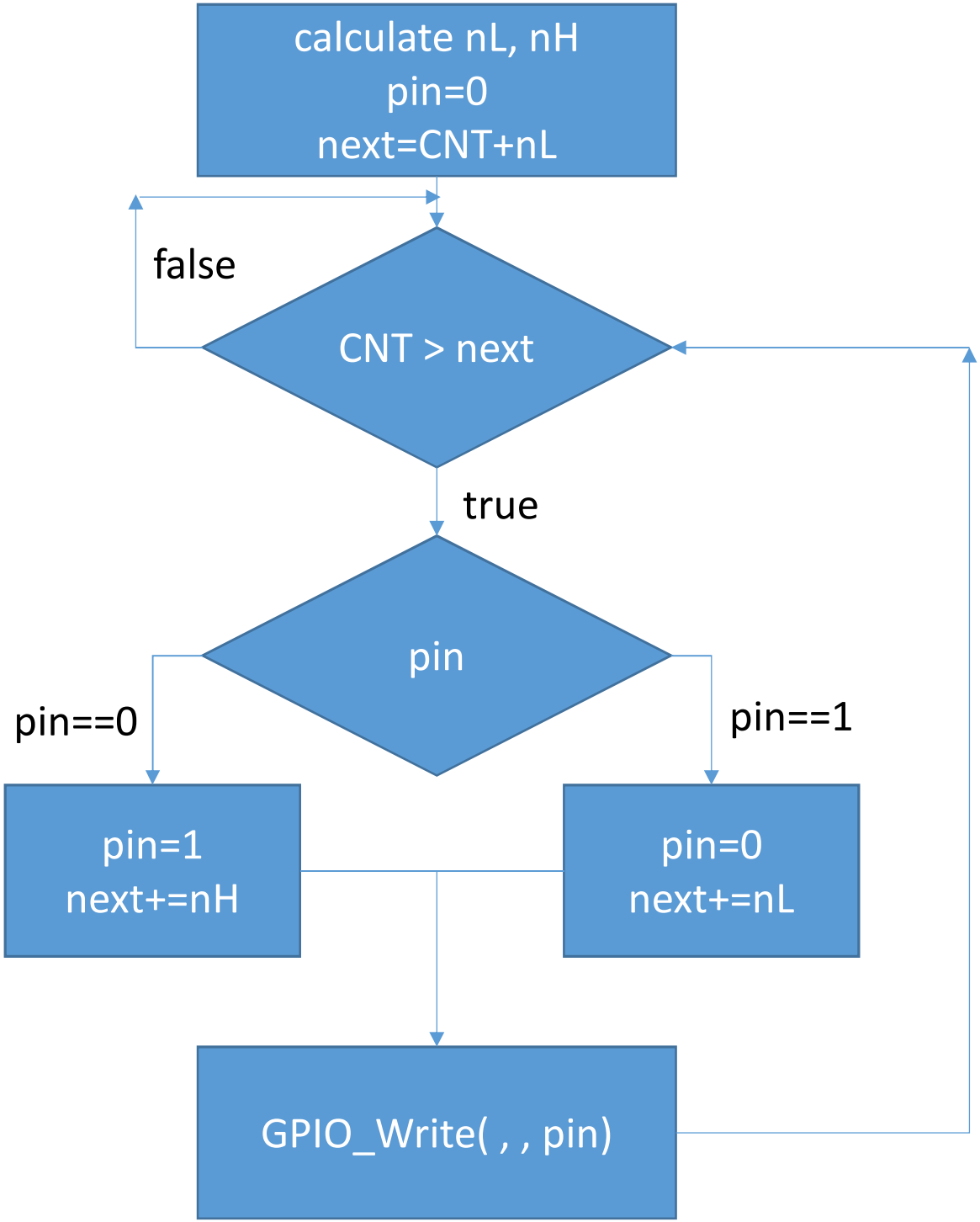
- Using Timer2(32-bit timer)
- TIMx_CNT - timer counter - is clocked by the prescaler output
- TIMx_PSC - prescaler counter - will divide source clock frequency
- TIMx_CR1 - timer control 1 - writing TIM_CR1_CEN to enable counter


Internal clock (CK_INT)
TIMxCLK from RCC
Trigger controller
ETRF
TRGO
to other timers
to DAC/ADC
TIMx_ETR
ETR
Polarity selection & edge detector & prescaler
ETRP
Input filter
ITR0
ITR1
ITR2
ITR3
ITR
TRC
TGI
TRGI
Slave controller mode
Reset, enable, up, count
TI1F_ED
TI1FP1
TI2FP2
Encoder interface
U
Auto-reload register
UI
Stop, clear or up/down
CK_PSC
PSC prescaler
CK_CNT
+/-
CNT counter
XOR
TI1
TI2
TI3
TI4
Input filter & edge detector
TI1FP1
TI1FP2
TI2FP1
TI2FP2
TI3FP3
TI3FP4
TI4FP3
TI4FP4
IC1
IC2
IC3
IC4
Prescaler
IC1PS
IC2PS
IC3PS
IC4PS
CC1I
CC2I
CC3I
CC4I
Capture/Compare 1 register
Capture/Compare 2 register
Capture/Compare 3 register
Capture/Compare 4 register
OC1REF
OC2REF
OC3REF
OC4REF
Output control
OC1
OC2
OC3
OC4
TIMx_CH1
TIMx_CH2
TIMx_CH3
TIMx_CH4

Square wave



Generate square wave



Exercise

Main program

- If button pressed, constantly generate 5Hz, 50% duty signal to LED. If button released, LED turns off.
- When LED turns off, timer should not count.
- Use your own driver.

RCC interface

- AHB Peripheral clocks configuration
  - void RCC_AHBPeriphClockCmd(uint32_t RCC_Periph, FunctionalState NewState);
- APB1 Peripheral clocks configuration
  - void RCC_APB1PeriphClockCmd(uint32_t RCC_Periph, FunctionalState NewState);

Timer interface

- Set prescaler value
  - void TIM_PrescalerConfig(TIM_TypeDef* TIMx, uint16_t Prescaler);
- Set counter value
  - void TIM_SetCounter(TIM_TypeDef* TIMx, uint32_t Counter);
- Get counter value
  - uint32_t TIM_GetCounter(TIM_TypeDef* TIMx);
- Enable/Disable counter
  - void TIM_Cmd(TIM_TypeDef* TIMx, FunctionalState NewState);