Timers III

ECE 3710

When everything is coming your way, you're in the wrong lane.

- Steven Wright

ex: 500ms delay with SysClk = 8 MHz

max delay w/16-bit counter
so we'll need to loop

strategy:

0. 1/8e6*2^16 ~ 8 ms << 500 ms
1. create 5ms delay
2. loop 100 times

use periodic timer
and counter overflows

ex: 500ms delay with SysClk = 8 MHz

strategy:

1. get 5 ms
2. run 100 times

$$(INITIAL + 1) \times \frac{1}{8\text{e}6}$$

$$n \times 0.125\mu\text{s} = 0.005 \rightarrow n = 40,000$$

$$INITIAL + 1 = 40,000 \rightarrow INITIAL = 39,999$$

0x9C3F

ex: 500ms delay with SysClk = 8 MHz

```
;assume timer is configured as above
  ldr R1,=TM0
  mov R2,#0 ;use R2 as counter
  mov R3,#1 ;for reseting expiry flag
  mov R4,#0 ;for stopping counter
wait ldr R0,[R1,#0x1C] ;timer status (GPTMRIS)
  ands R0,#0x1         ;sets Z=1 if R0==0
  beq wait             ;branch if Z=1
  ;clear timer expiry flag
  str R3,[R1,#0x24]
  add R2,#1 ;increment counter
  cmp R2,#100
  bne wait
  ;stop timer
  str R4,[R1,#0xC]
  ;clear timer expiry flag
  str R3,[R1,#0x24]
```

note: with such large delay, can ignore the odd machine cycle timer decrements at SysClk:

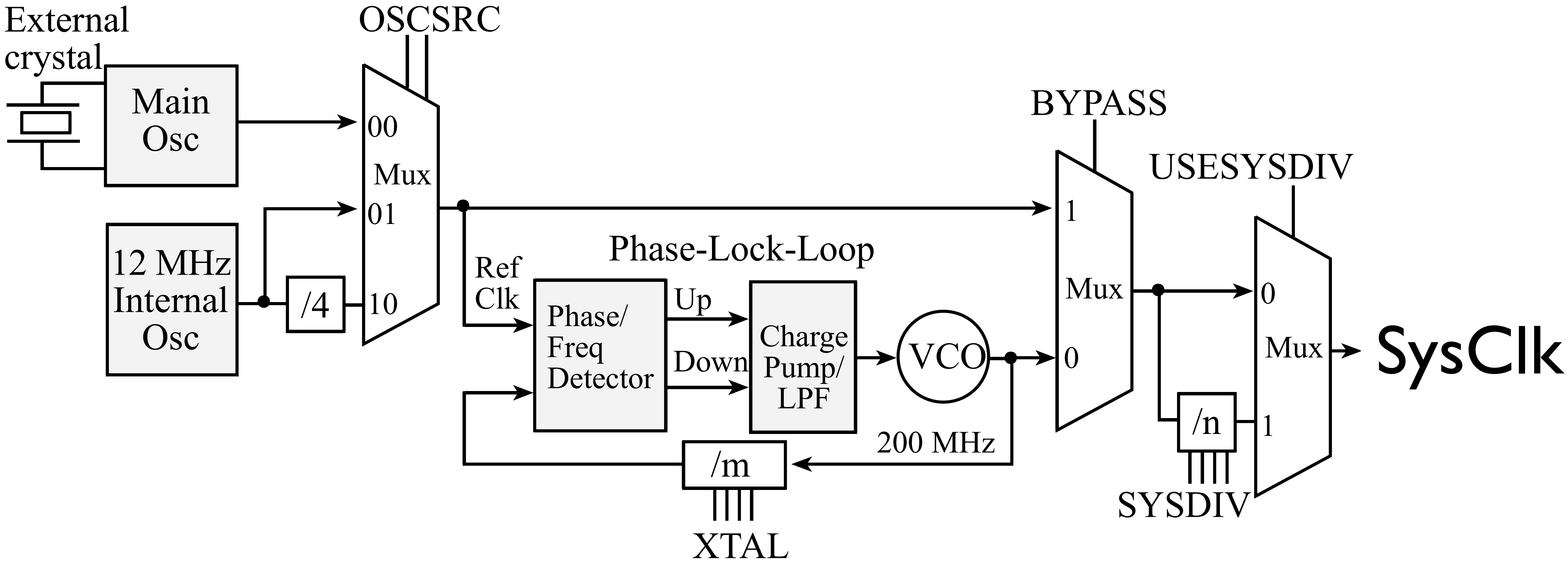
1 MC = 0.083 uS = 12.05 MHz

from simulator

so that doesn’t work?

never fear...

Frequency of SysClk?

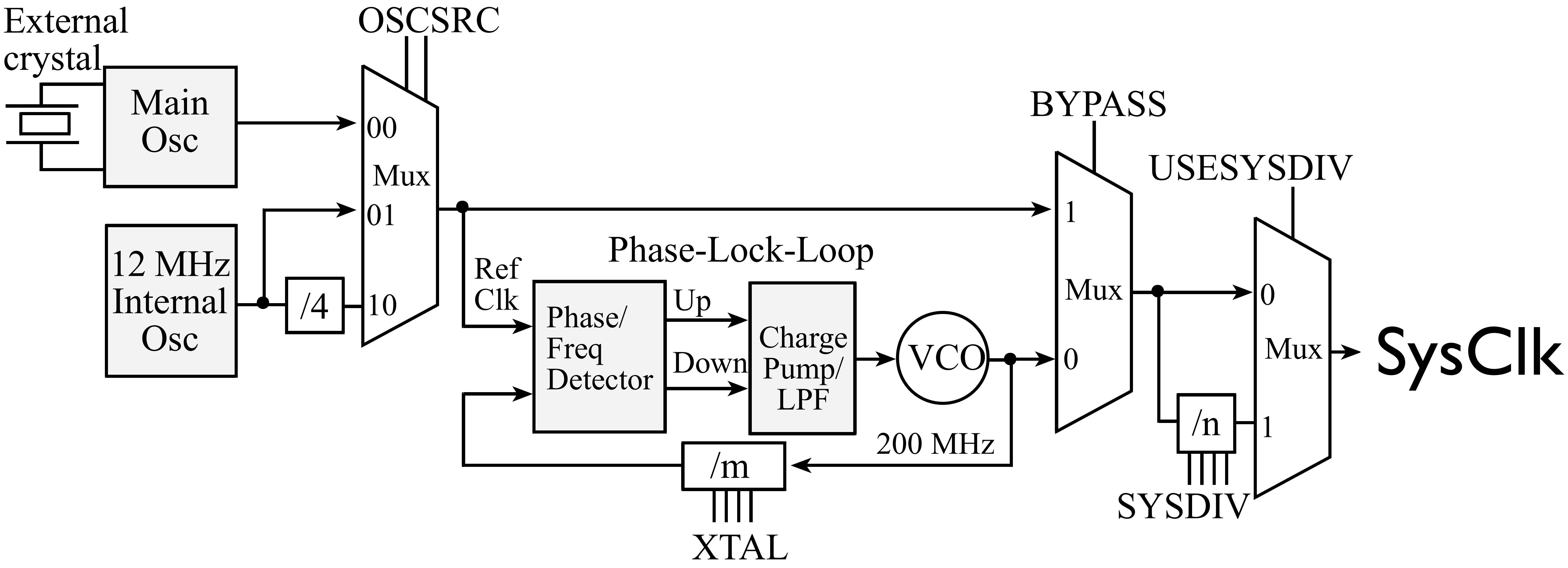


defaults:

```
OSCSRC=0b01
BYPASS=0b1
USESYSDIV=0b0
```

⟶ SysClk = 12 MHz (+/-30%)
Internal Osc

Frequency of SysClk?



clocks:

1. internal osc (12 MHz +/-30%)
2. internal osc (30 KHz +/-50%; not pictured)
3. main osc (a 1--8.192 MHz ext. crystal [some restrictions])
4. external RTC osc (not pictured)

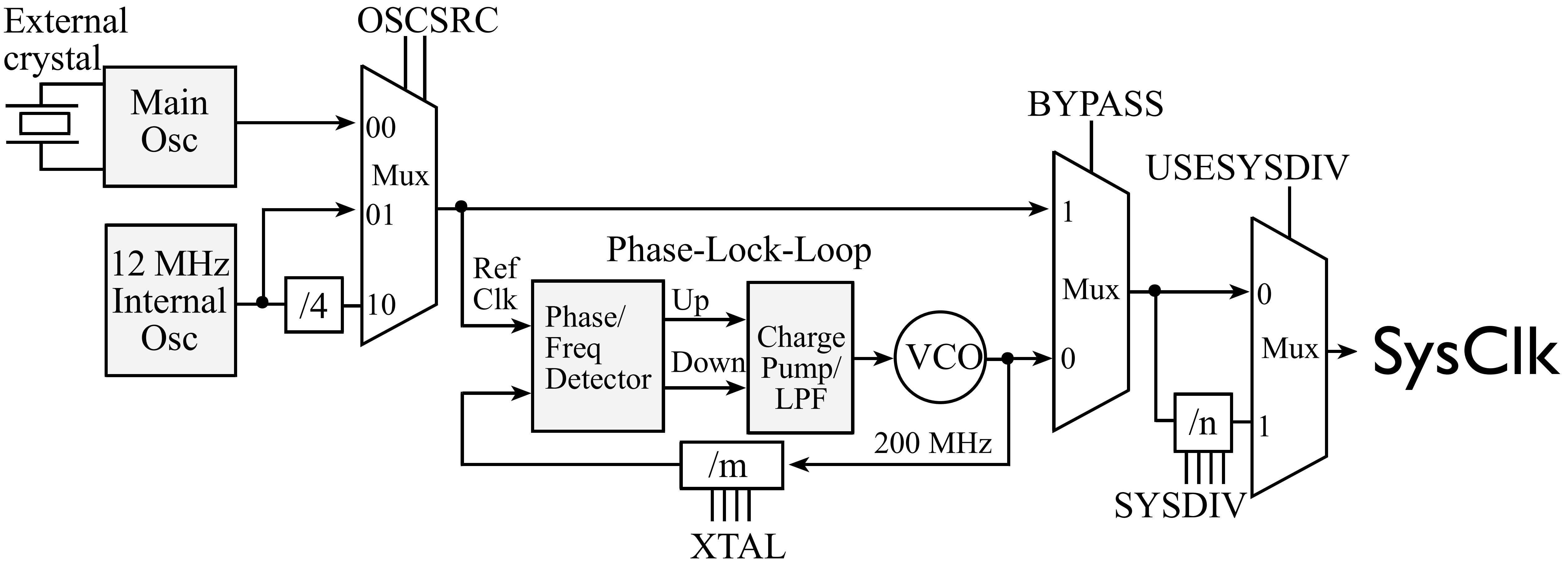
low power → 1

power saving → 2

high accuracy → 3

dual use → 4

Frequency of SysClk?



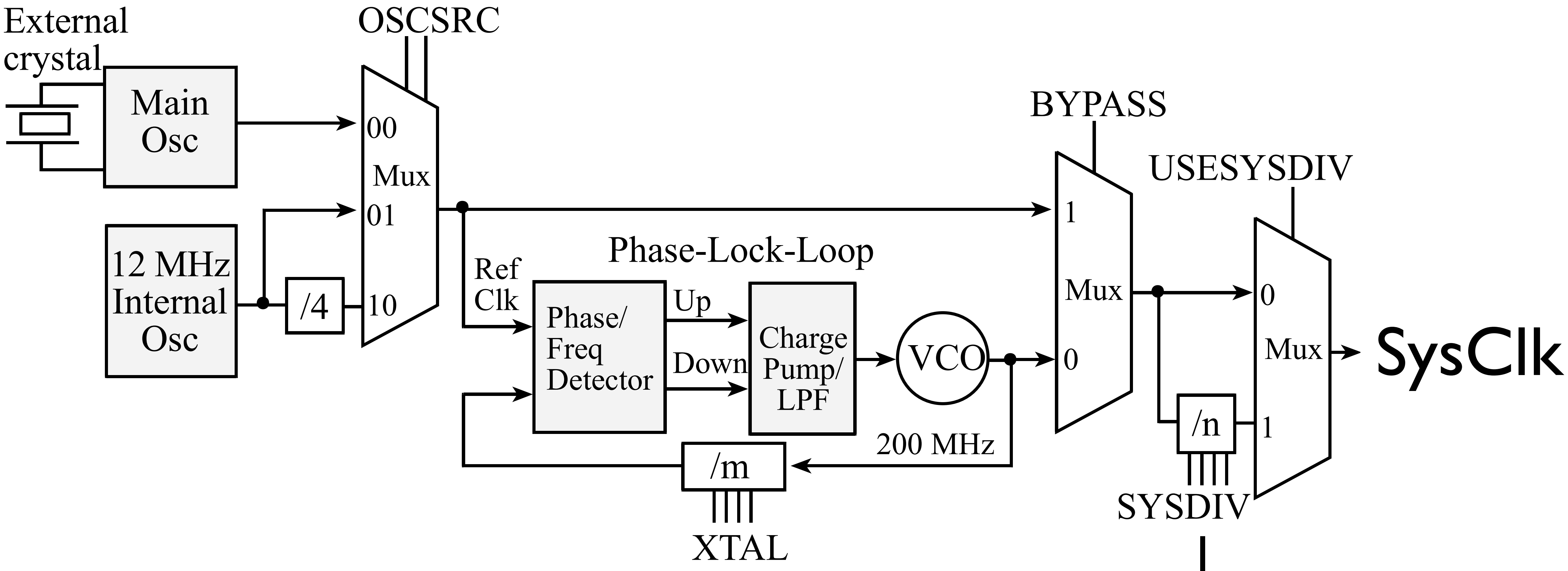
SysClk sources:

1. one to four above
2. internal osc/4 (3 MHz+/-30%)
3. PLL: 200 MHz (for SysClk; 400 MHz for CAN)

needs a reference source

(provided by main osc)

Frequency of SysClk?



0...15

SysClk speed:

1. nominal speed of sources
2. nominal speed/`SYSDIV`

e.g. using PLL (200 MHz)
w/divisor=8
=> SysClk=200/8=25 MHz

SysClk Config:

1. enable/disable Internal/Main osc
(IOSCDIS/MOSCDIS)

2. select osc source
(OSCSRC)

3. if external crystal is used, set XTAL
(XTAL)

4. select PLL or external/internal osc
(BYPASS)

5a. select frequency division or not
(USESYSDIV)

5b. set frequency divider
(SYSDIV)

Run-Mode Clock Configuration (RCC)

Base 0x400F.E000
Offset 0x060
Type R/W, reset 0x078E.3AD1

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved				ACG	SYSDIV				USESYSDIV	reserved	USEPWMDIV	PWMDIV			reserved
Type	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	RO	R/W	R/W	R/W	R/W	RO
Reset	0	0	0	0	0	1	1	1	1	0	0	0	1	1	1	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved		PWRDN	reserved	BYPASS	reserved	XTAL				OSCSRC		reserved		IOSCDIS	MOSCDIS
Type	RO	RO	R/W	RO	R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	R/W	R/W
Reset	0	0	1	1	1	0	1	0	1	1	0	1	0	0	0	1

p193

SysClk Config:

1. enable/disable Internal/Main osc
(IOSCDIS/MOSCDIS)

2. select osc source
(OSCSRC)

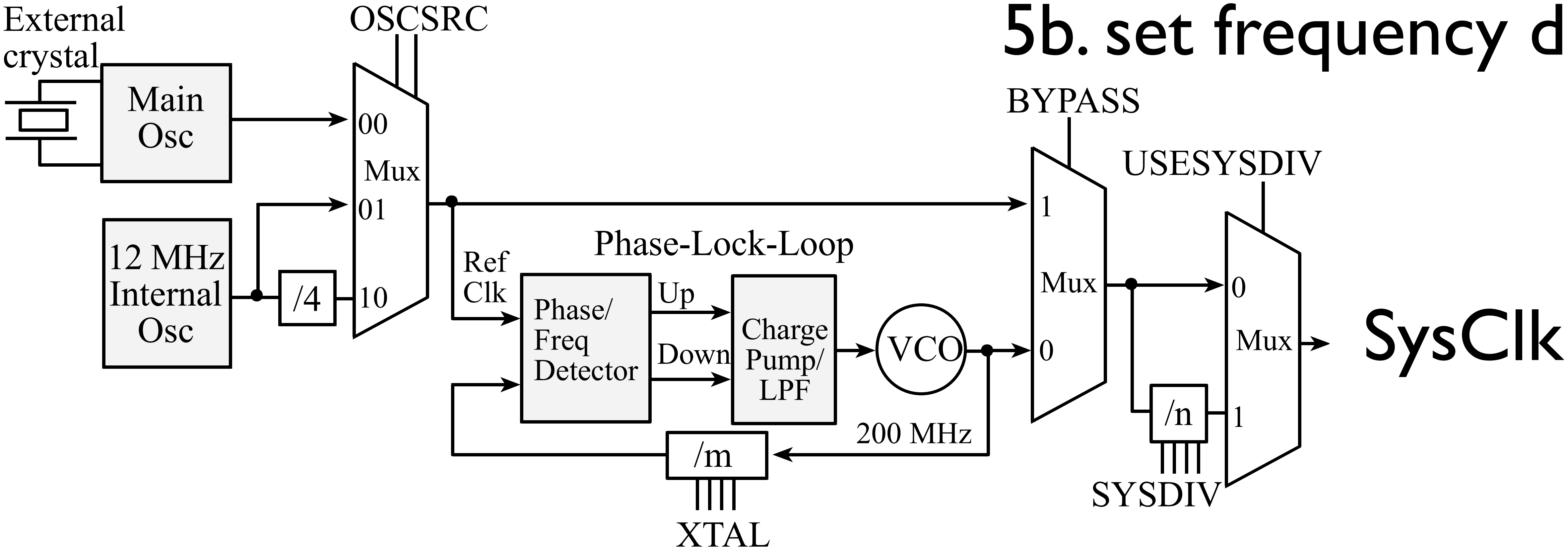
3. if external crystal is used, set XTAL
(XTAL)

4. select PLL or external/internal osc
(BYPASS)

5a. select frequency division or not
(USESYSDIV)

5b. set frequency divider
(SYSDIV)

see how this corresponds to figure



p193

OSCSRC Value

Value	Input Source
0x0	MOSC Main oscillator
0x1	IOSC Internal oscillator (default)
0x2	IOSC/4 Internal oscillator / 4
0x3	30 kHz 30-KHz internal oscillator

p195

XTAL values

Value	Crystal Frequency (MHz) Not Using the PLL	Crystal Frequency (MHz) Using the PLL
0x0	1.000	reserved
0x1	1.8432	reserved
0x2	2.000	reserved
0x3	2.4576	reserved
0x4	3.579545 MHz	
0x5	3.6864 MHz	
0x6	4 MHz	
0x7	4.096 MHz	
0x8	4.9152 MHz	
0x9	5 MHz	
0xA	5.12 MHz	
0xB	6 MHz (reset value)	
0xC	6.144 MHz	
0xD	7.3728 MHz	
0xE	8 MHz	
0xF	8.192 MHz	

can only use crystals w/these values

p195

SYSDIV values

use PLL

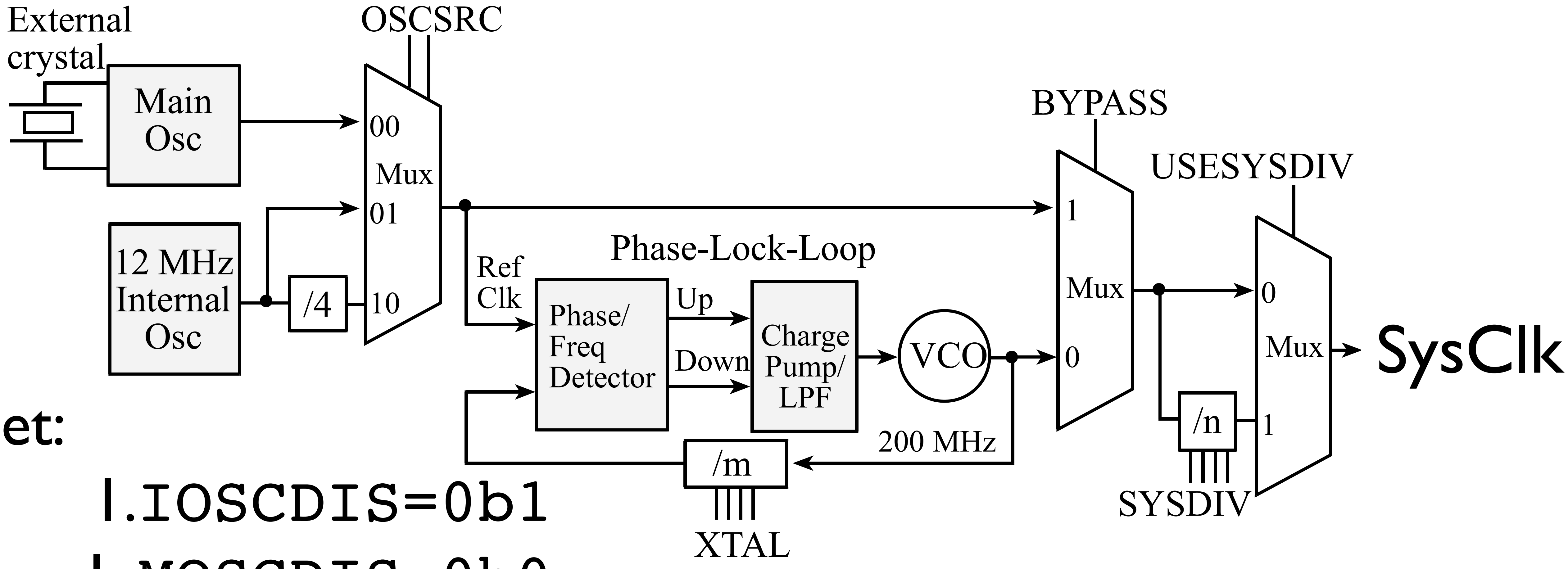
don’t use PLL

SYSDIV	Divisor	Frequency (BYPASS=0)	Frequency (BYPASS=1)
0x0	/1	reserved	Clock source frequency/2
0x1	/2	reserved	Clock source frequency/2
0x2	/3	reserved	Clock source frequency/3
0x3	/4	50 MHz	Clock source frequency/4
0x4	/5	40 MHz	Clock source frequency/5
0x5	/6	33.33 MHz	Clock source frequency/6
0x6	/7	28.57 MHz	Clock source frequency/7
0x7	/8	25 MHz	Clock source frequency/8
0x8	/9	22.22 MHz	Clock source frequency/9
0x9	/10	20 MHz	Clock source frequency/10
0xA	/11	18.18 MHz	Clock source frequency/11
0xB	/12	16.67 MHz	Clock source frequency/12
0xC	/13	15.38 MHz	Clock source frequency/13
0xD	/14	14.29 MHz	Clock source frequency/14
0xE	/15	13.33 MHz	Clock source frequency/15
0xF	/16	12.5 MHz (default)	Clock source frequency/16

note: `SYSDIV` = divisor-1

p179

ex: 8 MHz SysClk using ext. crystal (XTAL) = 8 MHz



set:

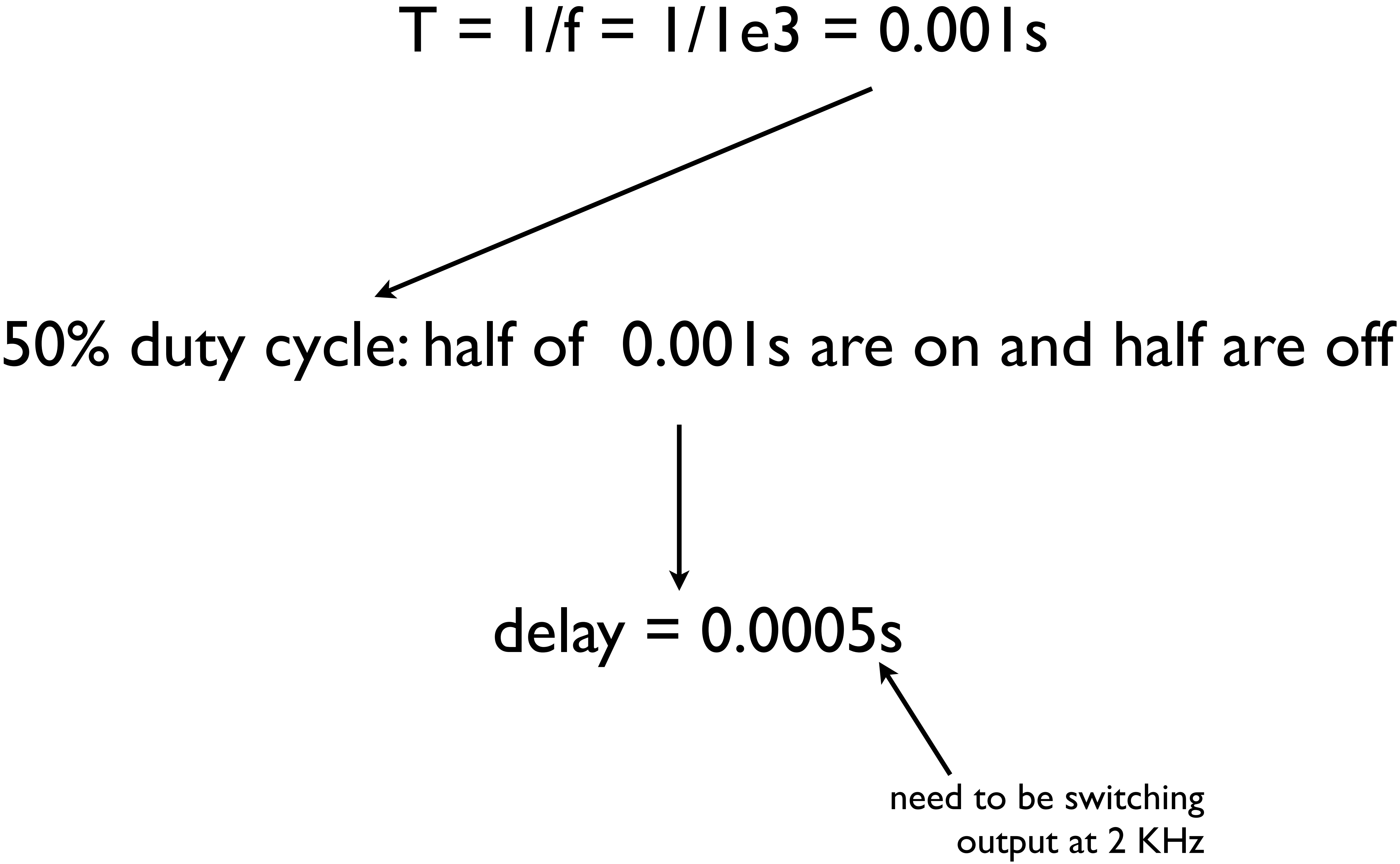
1. IOSCDIS=0b1
1. MOSCDIS=0b0
2. OSCSRC=0b00
4. BYPASS=0b1
5a. USESYSDIV=0b0
3. XTAL=0xE

→ SysClk = Main Osc

Q: 8 MHz SysClk using ext. crystal (XTAL) = 8 MHz for Tiva C?

ex: 50% duty cycle at 1 KHz (SysClk=12 MHz)

ex: 50% duty cycle at 1 KHz (SysClk=12 MHz)



T = 1/f = 1/1e3 = 0.001s

50% duty cycle: half of 0.001s are on and half are off

delay = 0.0005s

need to be switching
output at 2 KHz

ex: 50% duty cycle at 1 KHz (SysClk=12 MHz)

delay = 0.0005s:

$$(INITIAL + 1) \times \frac{1}{12\text{e}6}$$

$$n \times 0.083\mu\text{s} = 0.0005\text{s} \rightarrow n = 6000$$

$$INITIAL + 1 = 6,000 \rightarrow INITIAL = 5,999$$

0x176F

ex: 50% duty cycle at 1 KHz (SysClk=12 MHz)

4. select periodic

```
ldr R1,=TM0 ;timer0 base
mov R0,#0x2
str R0,[R1,#0x4]
```

GPTMTAMR

Offset 0x004
Type R/W, reset 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved												TAAMS	TACMR	TAMR	
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0x0	Reserved
0x1	One-Shot Timer mode
0x2	Periodic Timer mode
0x3	Capture mode

set to 0x2

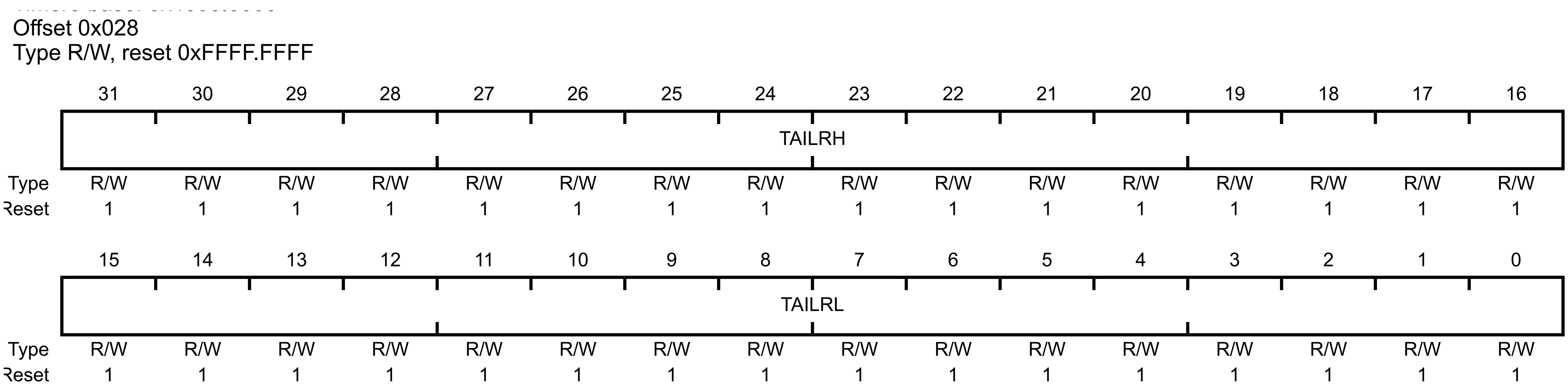
p346

ex: 50% duty cycle at 1 KHz (SysClk=12 MHz)

5. set initial value

```
ldr R1,=TM0 ;timer0 base
ldr R0,=0x176F
str R0,[R1,#0x28]
```

GPTMTAILR



Offset 0x028
Type R/W, reset 0xFFFF.FFFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TAILRH															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAILRL															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

can only use lower 16 bits

p359

ex: 50% duty cycle at 1 KHz (SysClk=12 MHz)

```
;assume timer is configured as above (PD0, too)
  ldr R1,=TM0
  mov R3,#1 ;for reseting expiry flag
  ldr R5,=PD0_DATA_B
wait ldr R0,[R1,#0x1C] ;timer status (GPTMRIS)
  ands R0,#0x1         ;sets Z=1 if R0==0
  beq wait             ;branch if Z=1
  ;clear timer expiry flag
  str R3,[R1,#0x24]
  ;flip pin
  ldr R2,[R5] ;R2=PD0
  mvn R2,R2   ;R2=~R2
  str R2,[R5] ;PD0=~PD0
  b wait
```

note: with such large delay, can ignore the odd machine cycle finally done with timers...

...aren't we?

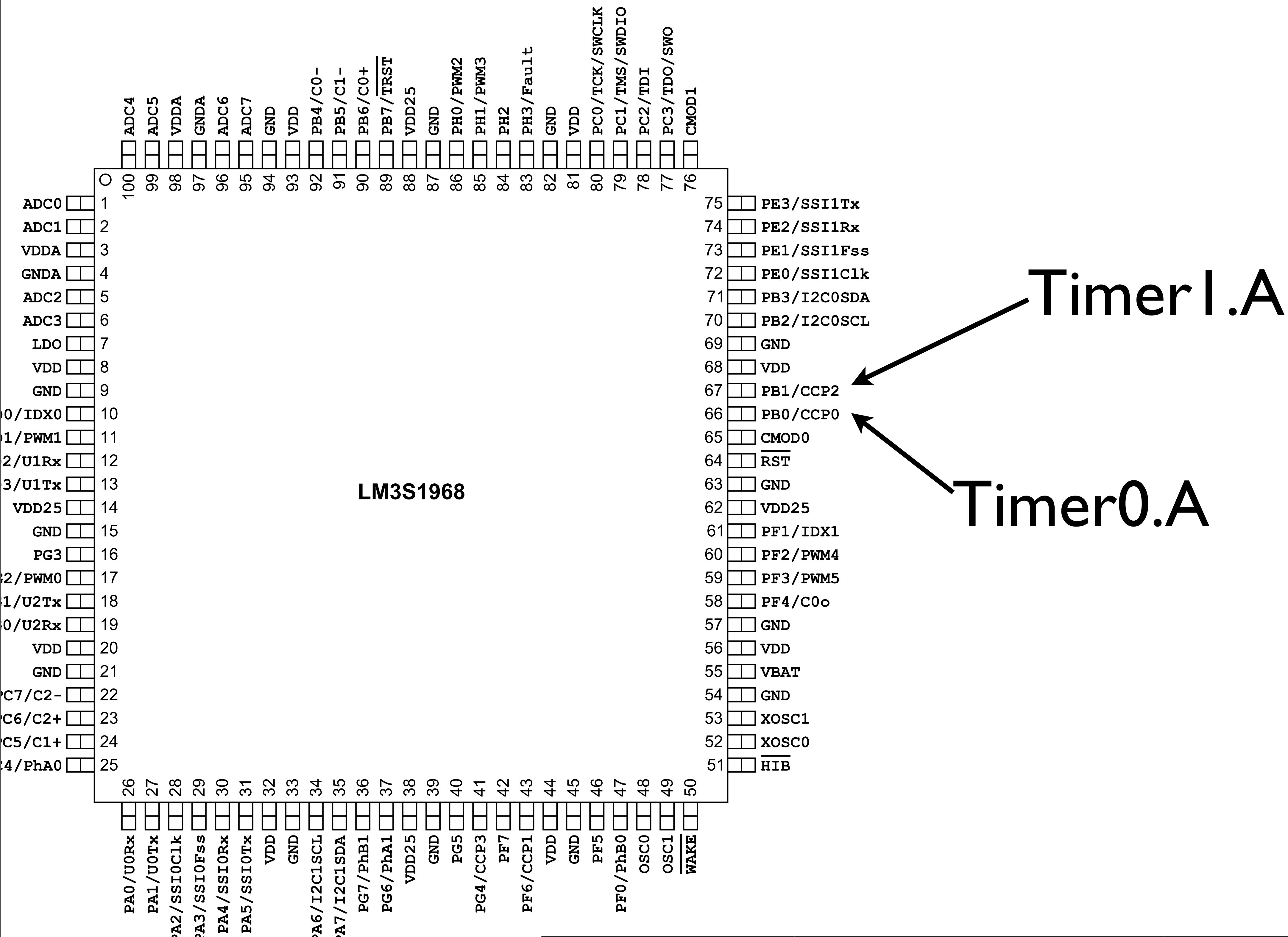
this is what
gets counted

timer (counter) programming:

1. internal (internal clock)
2. external (events)

will use these
interchangeably

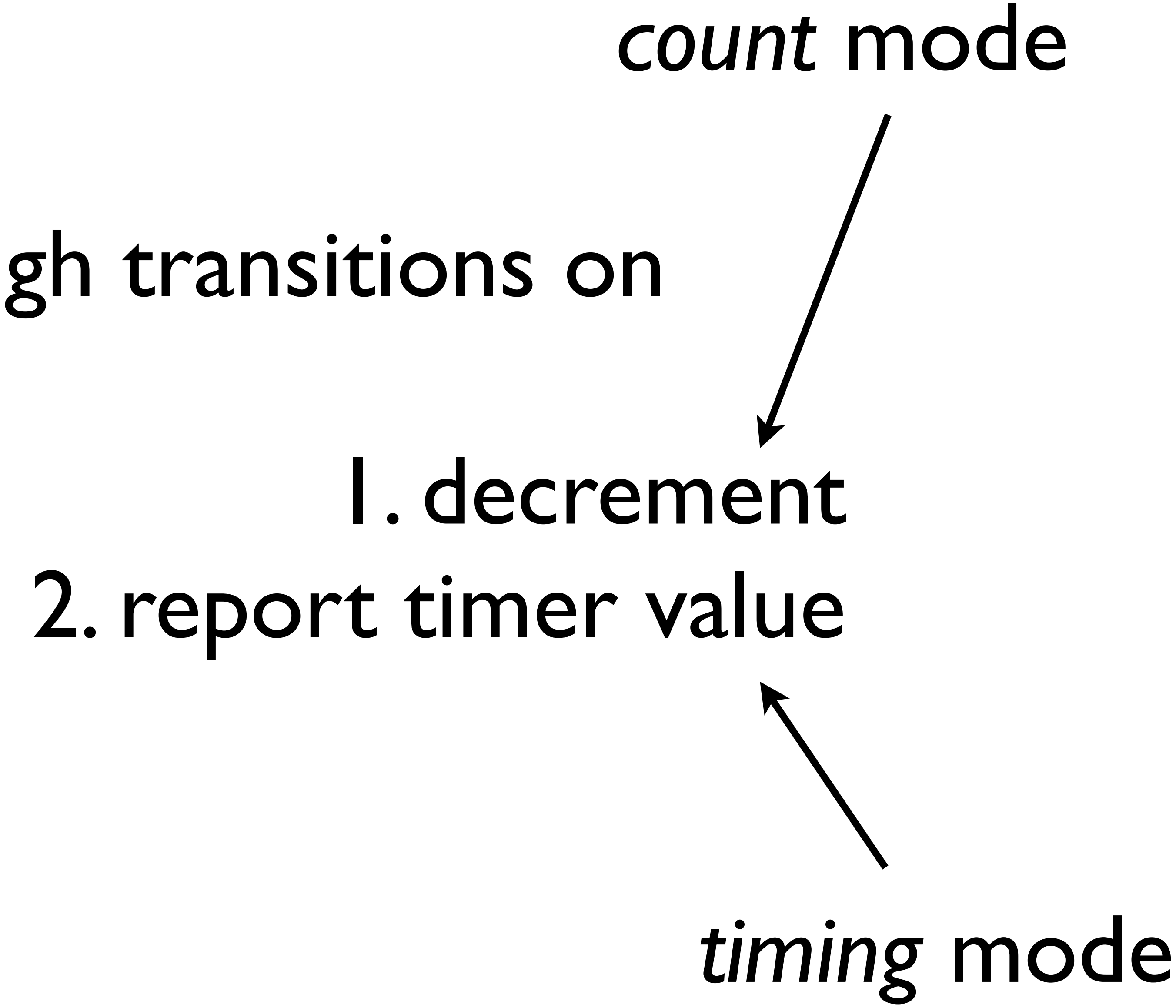
Q: which ports to connect external to?



Timer	16-Bit Up/Down Counter	Even CCP Pin	Odd CCP Pin
Timer 0	TimerA	CCP0	-
	TimerB	-	CCP1
Timer 1	TimerA	CCP2	-
	TimerB	-	CCP3

external timer modes

when high2low/low2high transitions on timer port:



count mode

1. decrement
2. report timer value

timing mode