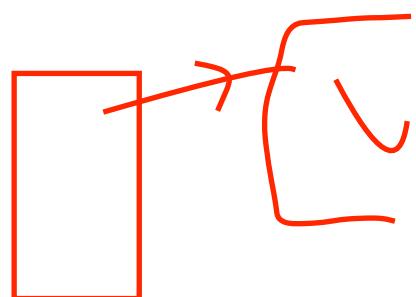


Clock Module

- It configures the system clocks (MCLK, SCLK, ACLK, ~~HSE/HSE~~ ...)
- Manages & meets the conflicting requirements of:
 - low power consumption
 - high frequency for fast response
 - accuracy & stability

Types of Clock Signals (Recap)

- RC (Resistor-Capacitor): built-in, lowest power consumption, see data sheet for accuracy & drift
- Crystal: Oscillator circuitry in chip, crystal & capacitors soldered on the board (e.g. 6 pF capacitors w/ 32 kHz)
- Ceramic resonator: like the crystal
- Bypass: An external clock signal connected to a pin on the MCU.



Clock Modules in MSP430 Chips

28

- x2xx Family : Basic Clock Module + (BCM+)
- x4xx Family : Frequency Locked Loop + (FLL+)
- FRGxx Family: Clock System (CS)
 - Lab board
 - well talk about this

Six Clock Sources

1) LFXT : Low-Frequency Crystal Oscillator

- Crystal up to 50 kHz
- Crystal (w/ capacitors) connected to LFXIN, LFXOUT pins.
- Internal capacitors may be used (e.g. 1, 2, 6, 10.5 pF)
- Pins should be diverted to LFXIN, LFXOUT functionality & fault flags should be cleared.

config_ACLK_to_32kHz_crystal();

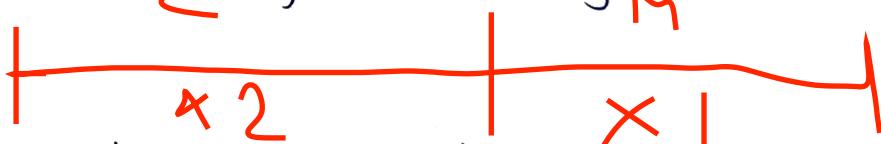
(2) HFXT CLK:

- HFXT: High-Frequency Crystal Oscillator
- Accepts a crystal up to 24 MHz
(HSP430 CPU runs at up to 16 MHz)

(3) VLOCLK

- VLO: Very Low-Frequency Oscillator
- RC, built-in
- Typical frequency: 9.4 kHz
- Range [6 - 14] kHz
- 0.2% / °C drift 0.7% / V drift
- Lowest power consuming clock signal.

(4) DCOCLK



- DCO: Digitally Controlled Oscillator
- RC, built-in, programmable to various frequencies
- Choices: 2.67, 3.5, 4, 5.3, 7, 8, 16 MHz
- Above 8 MHz, FRAM doesn't work (clock too fast)
- Activate wait state in the FRAM controller

✓ ✓

✓ ✓ ~

→ How does the CPU run at 1 MHz at startup?

(5) MODCLK

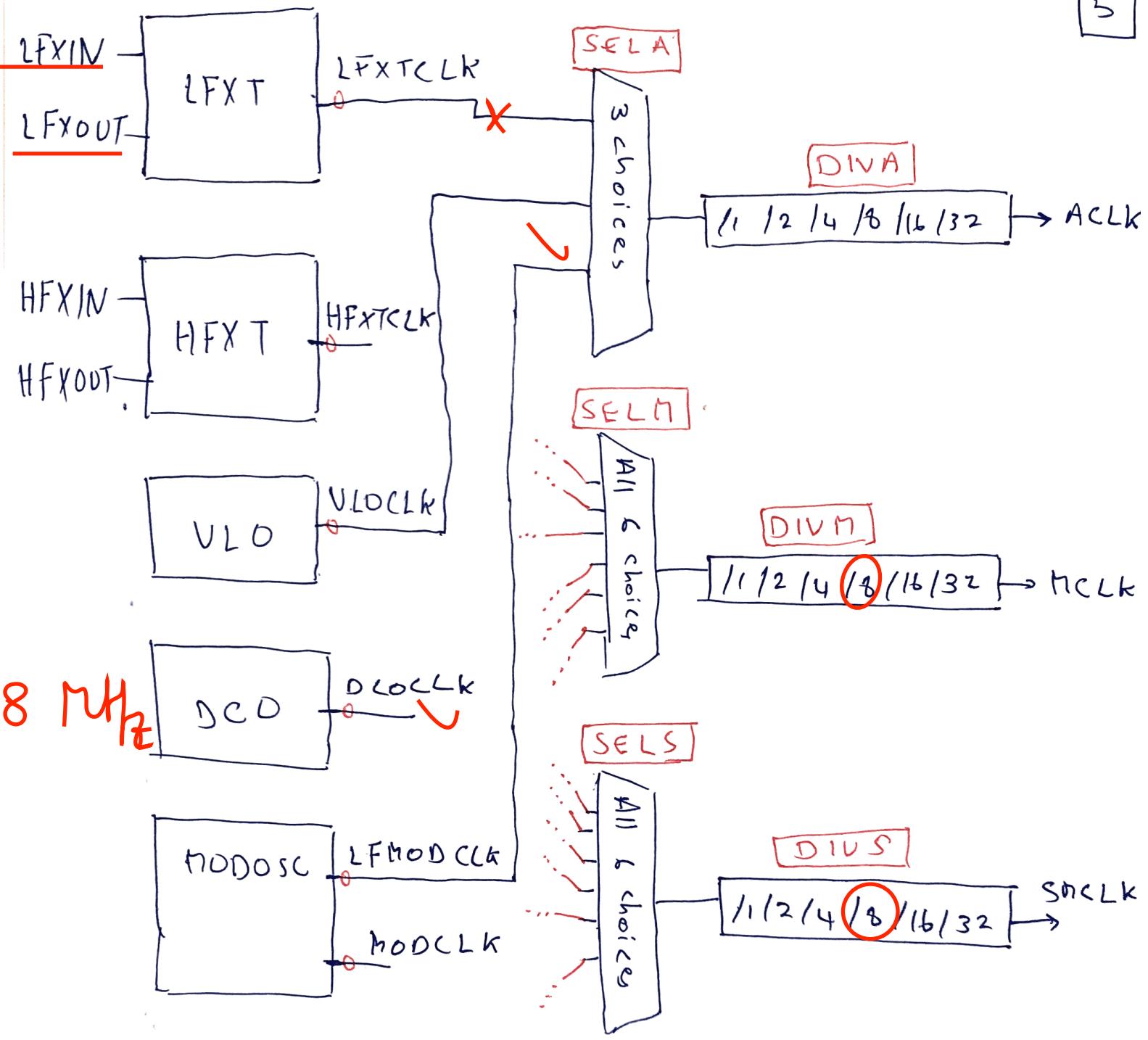
4

(6) LFMODCLK

- MOD: Module (MODOSC: Module Oscillator)
- LFMOD: Low-Frequency Module
- RC, built-in
- Typical: 4.8 MHz
- Range $[4 - 5.4] \text{ MHz}$
- $\text{LFMODCLK} = \text{MODCLK} / 128$
 - = $4.8 \text{ MHz} / 128$
 - = 37.5 kHz
 - A substitute for 32 kHz clock
 - Can be confused w/
 32 kHz clock!

System Clock Configuration

- ACLK can use 3 choices (LFXTCLK, VLOCLK, LFMODCLK)
- MCLK can use all 6 choices
- SCLK can use all 6 choices
- MODCLK & VLOCLK can be used directly by peripherals.



Clock System (CS) peripheral

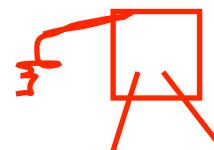


Where would we find the configurators
SELA, **DIVA**... ?

(to peripherals)

Crystal Recommendations

- The crystal is highly susceptible to noise.
- Placed close to the HCU.
- Placed away from High frequency lines
- Crystal housing is connected to Ground.
- See SLAA322B.



Default Configuration

- DCO running at 8 MHz
- MCLK & SMCLK choose DCO and use their divider = 8 (DIVM, DIVS)
- Therefore, $MCLK = SMCLK = 1 \text{ MHz}$ → default MCLK across all HSP43Q chips.
- ACLK from LFXT : 32 kHz crystal (Our board)
- But pins LFXIN & LFXOUT are GPIOs at startup
- Therefore, LFXT is considered failed : raises fault flags
- And ACLK diverts to LFMODCLK at 37.5 kHz.

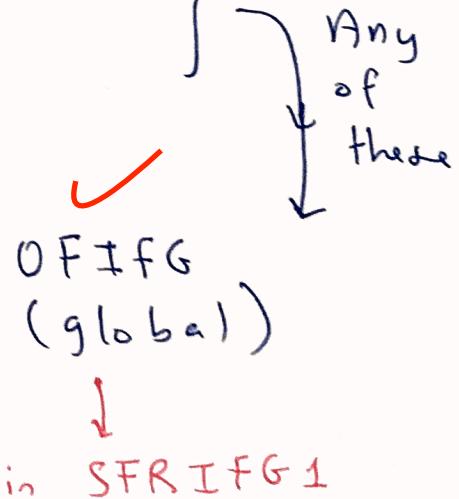
Overriding Feature

7

- The CS module supports this feature
- This feature is ON by default ; it can be turned OFF
- If a Low-Power mode shuts down a clock signal & a peripheral requests that signal, the clock turns ON for as long as it's requested.

Clock Failure Flags

- A flag is raised when a clock fails or is erratic.
- LFXT → LFXTOFFG (Oscillator Fault Flag)
- HFXT → HFXTOFFG
- OFIFG can raise an interrupt if enabled.
- Once raised, it remains ON until cleared by software.



```
config-ACLk-to-32-kHz-crystal () {  
    Divert pins to LFXIN, LFXOUT  
    while( LFXTOFFG == 1 OR OFIFG == 1 ) {  
        LFXTOFFG = 0;  
        OFIFG = 0;  
    }  
}
```

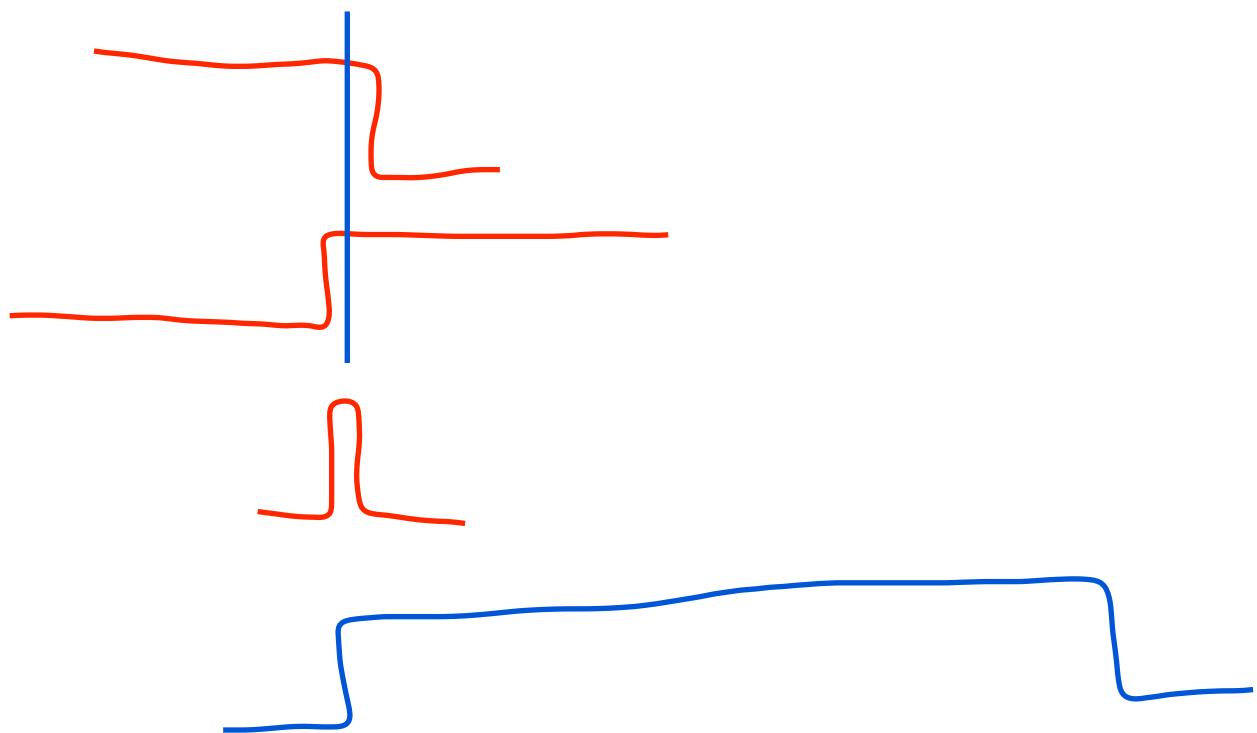
- Internal clocks are less likely to fail.

Fail-Safe Operation

- Crystal clocks are more likely to fail (crystal falls off, interference ...)
- The module automatically diverts to internal clock sources.
 - what could go wrong with the app. when this happens?

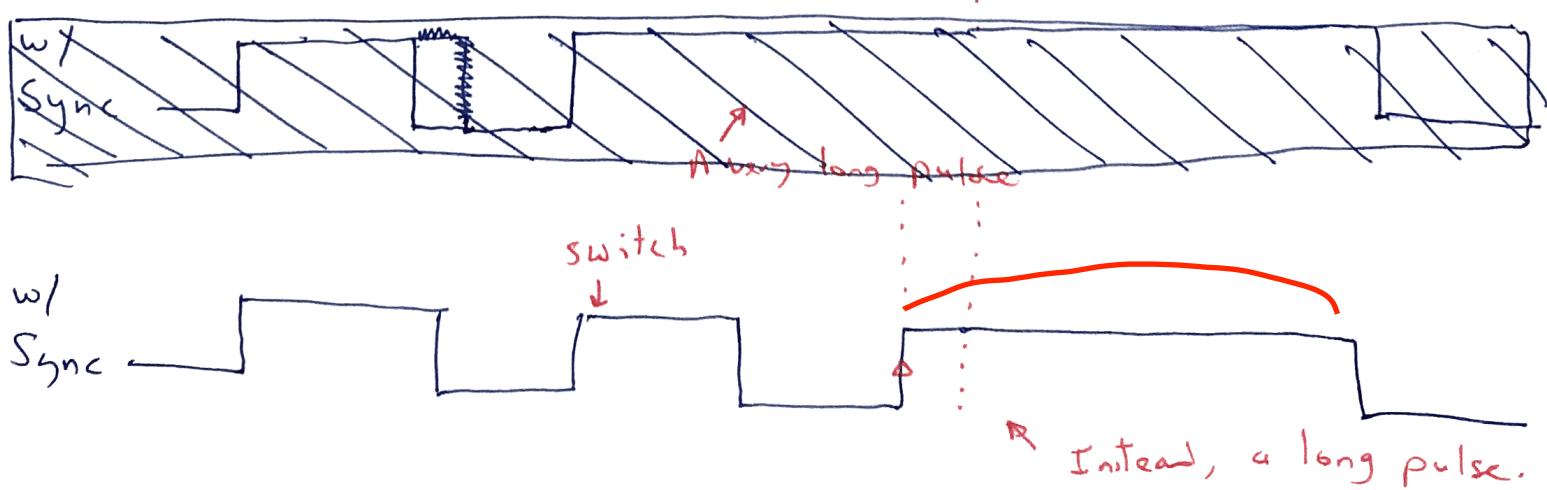
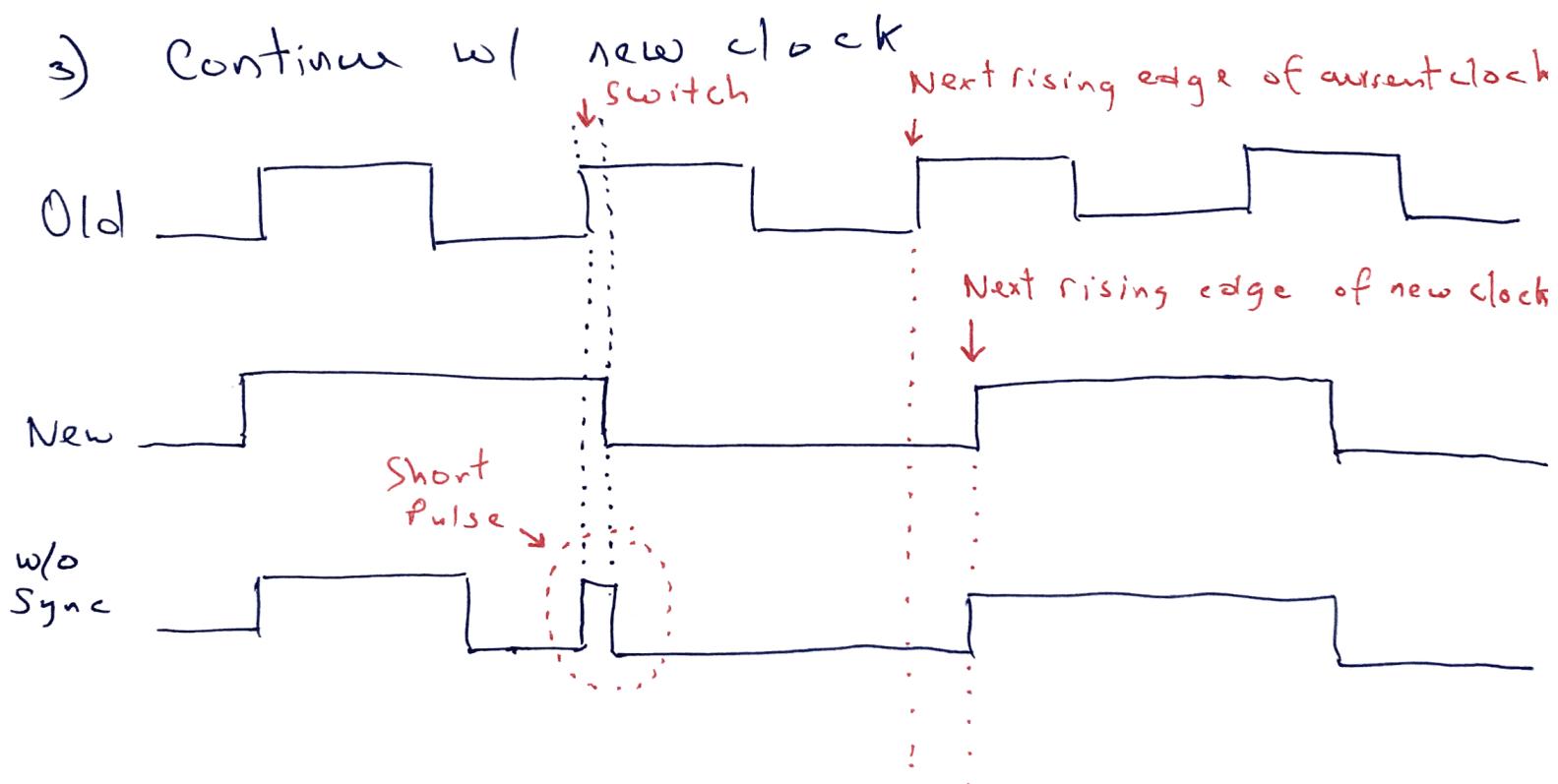
DCO Frequency Transition

- DCO can be configured by software at run-time
- After a frequency change, it's held off for 4 cycles so it becomes stable.



Synchronization when Changing the Clock Source 9

- Problem: Risk of getting a very short pulse
- Procedure:
 - 1) Current signal continues until next rising edge
 - 2) Clock remains high until next rising edge of new clock.
 - 3) Continue w/ new clock



- ④ The clock synchronization procedure is done by the clock module's hardware.
It's invisible to the programmer.

10

Clock System (CS) Configuration Registers

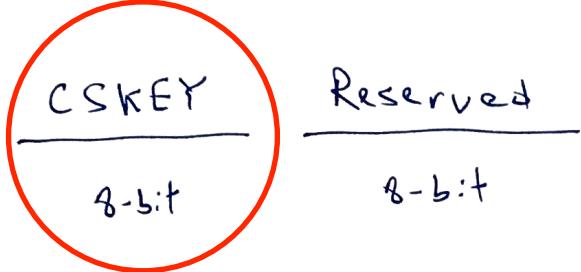
FR6XX Family User Guide - "Clock System" chapter
(end of chapter)

CSCTL0, CSCTL1, ..., CSCTL6
:
password protection 1, 2, 3 are the most useful

Password Protection

- Against unintentional access to config. registers (can fail the whole system if clocks stopped)
 - Protects access to all regs CSCTL1 to 6
- ④ How could the software unintentionally access CSCTLx registers?

CSCTLQ:
(16-bit)



CSKEY =
0xA500
in .h file
16-bit

1) Unlock before modifying the configuration

Write 16-bit CSKEY constant

to CSCTLQ

2) Relock when config. is done

Write a bad password to the upper byte
(as a byte access)

Eg.

CSCTLQ = CSKEY;

:

CSCTLQ - H = 0; // bad password

Assm:

MOV.W #CSKEY, & CSCTLQ

;

MOV.B #0, & CSCTLQ - H

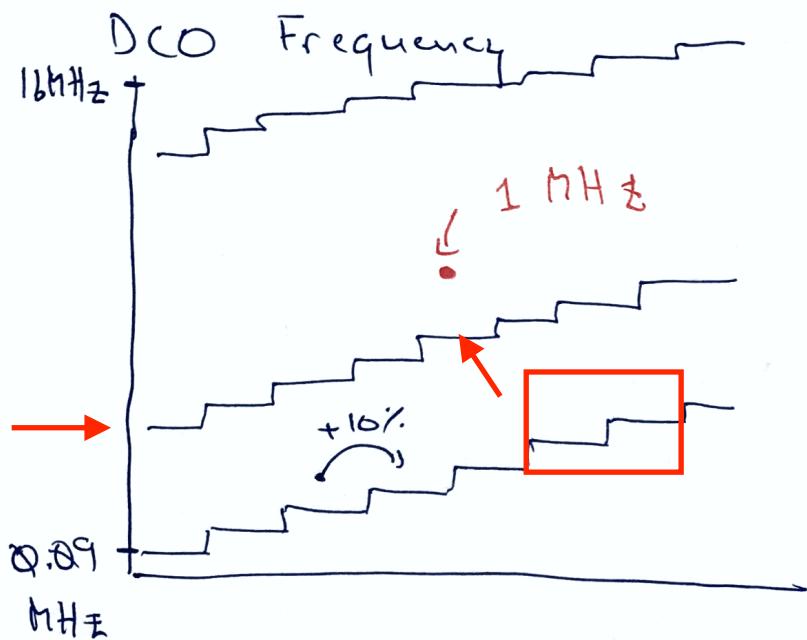
byte

Ex: Configure MCLK to 16 MHz (fastest CPU freq.) [12]

```
// DCO @ 16 MHz (DCOSEL = 1, DCOFSEL = 4),  
// DIVM = 2 (1)  
CSCTL0 = CSKEY; // Unlock CS Regs  
  
FRCTL0 = FRCTLPw | NWAITs-1;  
// Activate FRAM wait state  
// up to 7  
// beyond 8 MHz  
{  
CSCTL1 &= ~DCOFSEL-7; // 3-bit field  
CSCTL1 |= DCOFSEL-4;  
CSCTL1 |= DCORSEL-1;  
CSCTL3 &= ~(BIT2|BIT1|BIT0); // DIVM constants  
// not defined  
// in .h file  
CSCTL0_H = 0; // Lock CS Regs.
```

Basic Clock Module + (BCM+) Overview (x2xx)

13



Each line is called
a range
(16 ranges)

Each step is called a tap
(8 taps / range)

Default: Middle range, Middle tap on it

$$\Rightarrow 1.048576 \text{ MHz} \quad (2^{20})$$

Modulation

- Allows selecting a frequency that's between 2 taps
- Mixing cycles from 2 adjacent taps based on 32-cycle period.

E.g.: $F_1 = 0.95 \text{ MHz}$ $F_2 = 1.05 \text{ MHz}$ \swarrow Adjacent taps
 $\dots \dots \dots \dots \dots \dots \dots \dots$ add up to 32

$$\text{Result} = \frac{16 \times F_1 + 16 \times F_2}{32} = 1.00 \text{ MHz}$$

⇒ In Bch+,

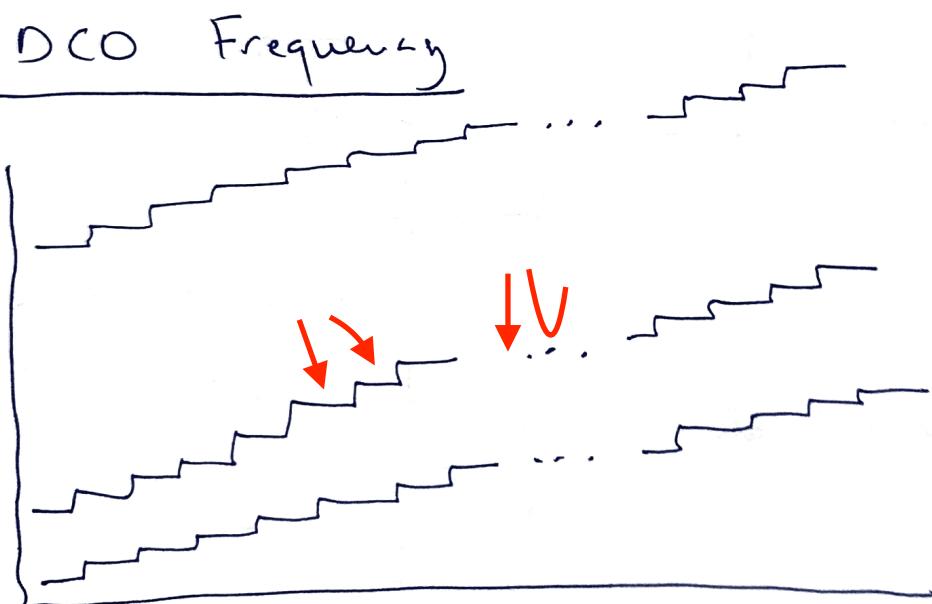
[14]

$$\text{DCO : MCLK : SHCLK} = \underline{1.448576 \text{ MHz}} \quad (\underline{\text{Raw}})$$

1.448576 MHz
(calibrated)

A few lines of code to do ^{the} modulation

Frequency Locked Loop + (FLL+) Overview (x4xx)



- 5 ranges (lines)
 - 28 taps (steps) per range
- why does the range have too many taps ?

Frequency Locked Loop

15

- Stabilize DCO by comparing its output to the crystal clock
- DCO frequency is specified as a multiple of crystal frequency.

$$\begin{aligned} F_{DCO} &= \boxed{32} \times \underline{F_{ACLK}} = 32 \times 32 \text{ kHz} \\ &= 1 \text{ MHz} \end{aligned}$$

