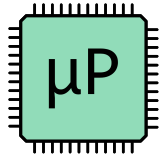
Common Digital Structures

Lecture 7

Microprocessor-based Systems (E155)

Prof. Josh Brake



Quiz!

1. When designing synchronous digital circuits…
   a) What is the setup time constraint?
   b) What is the hold time constraint?
   c) If we make a mistake with one of these constraints, can we fix it after building circuitry?
2. What are pull up and pull down resistors used for?
3. Do we have to worry about setup/hold time constraints when using our microcontroller?

https://docs.google.com/presentation/d/1ShVehgj6aX2Dr44j0UIh4JXsUJaZQ8HdsNg32T6HZcs/edit

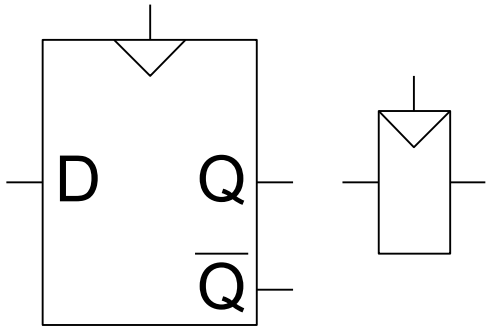
Outline

- Overview of Common Digital Structures
  - Flip Flops
  - Counters
  - Shift Registers
  - Memory arrays
    - RAM
    - Flash
- Synchronous digital design review
  - Setup and hold time constraints
- Pullup and pulldown resistors
- Lab 3 Questions

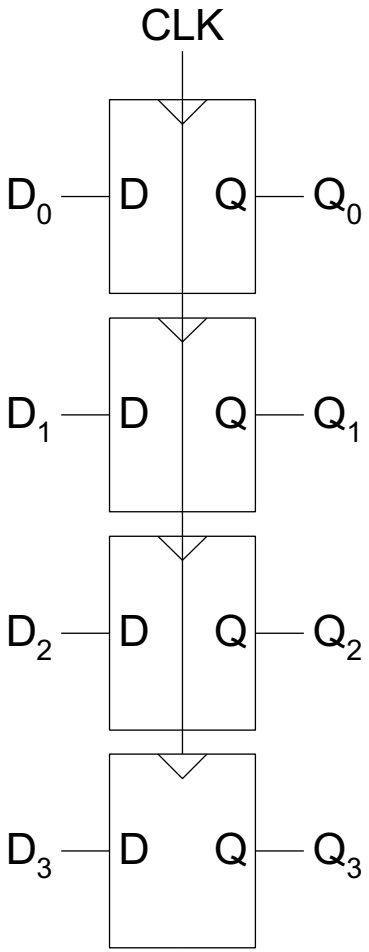
D Flip-Flop

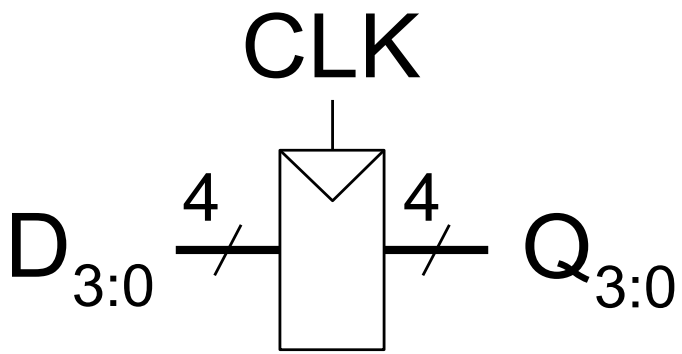
- **Inputs:** *CLK*, *D*
- **Function**
  - Samples *D* on rising edge of *CLK*
    - When *CLK* rises from 0 to 1, *D* passes through to *Q*
    - Otherwise, *Q* holds its previous value
  - *Q* changes only on rising edge of *CLK*
- Called *edge-triggered*
- Activated on the clock edge

D Flip-Flop Symbols



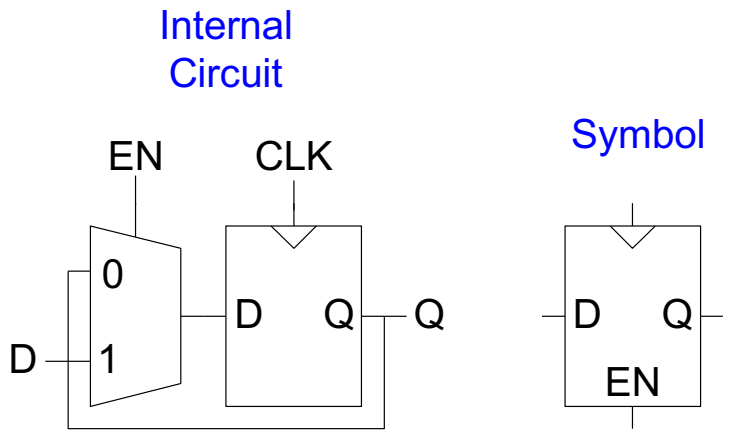
Registers: Multi-bit Flip-Flop





Enabled Flip-Flops

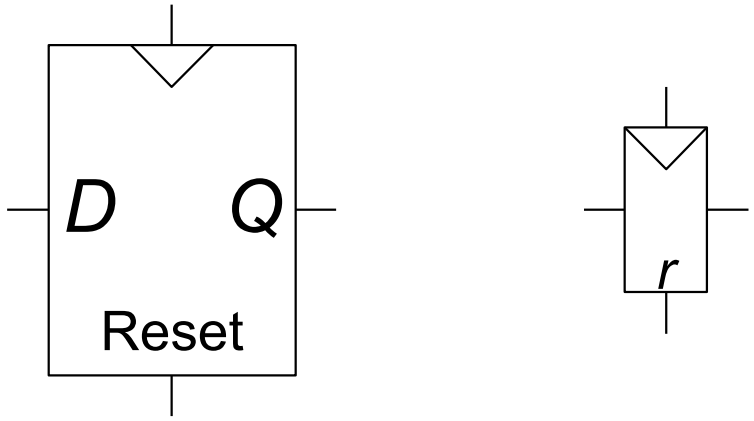
- **Inputs:** *CLK*, *D*, *EN*
  - The enable input (*EN*) controls when new data (*D*) is stored
- **Function**
  - ***EN* = 1:** *D* passes through to *Q* on the clock edge
  - ***EN* = 0:** the flip-flop retains its previous state



Resettable Flip-Flops

- **Inputs:** *CLK*, *D*, *Reset*
- **Function:**
  - ***Reset* = 1:** *Q* is forced to 0
  - ***Reset* = 0:** flip-flop behaves as ordinary D flip-flop

Symbols



Counters

- N-bit output with size of the bus
- Counter + Register with reset
- Counts on clock edge
  - 000, 001, 010, 011, 100, 101, 110, 111, 000, 001, …
- Where are these in your MCU
  - Clock divisors/prescalers
  - Timers

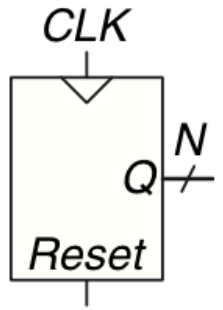


Figure 5.31 Counter symbol

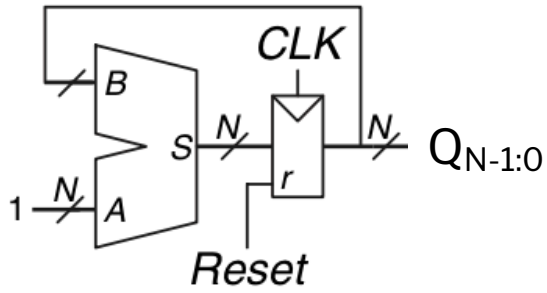
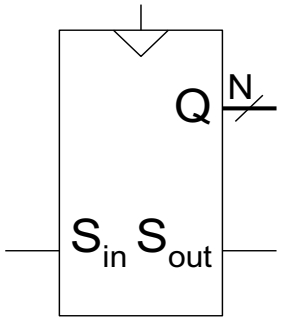


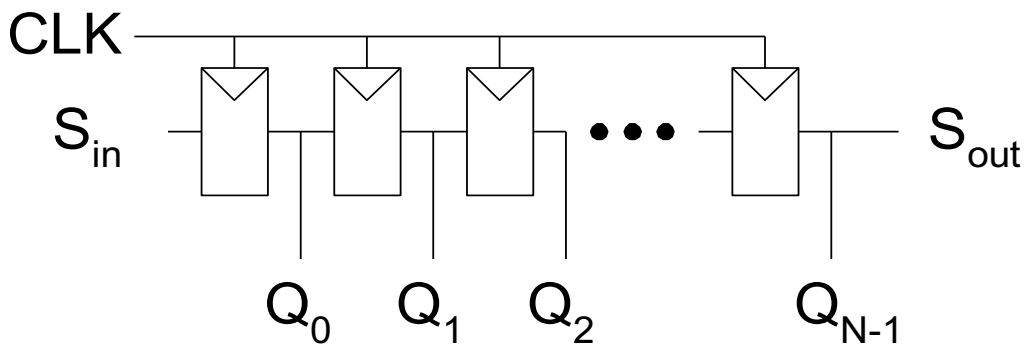
Figure 5.32 *N*-bit counter

DDCA ARMed p. 260

Shift Register

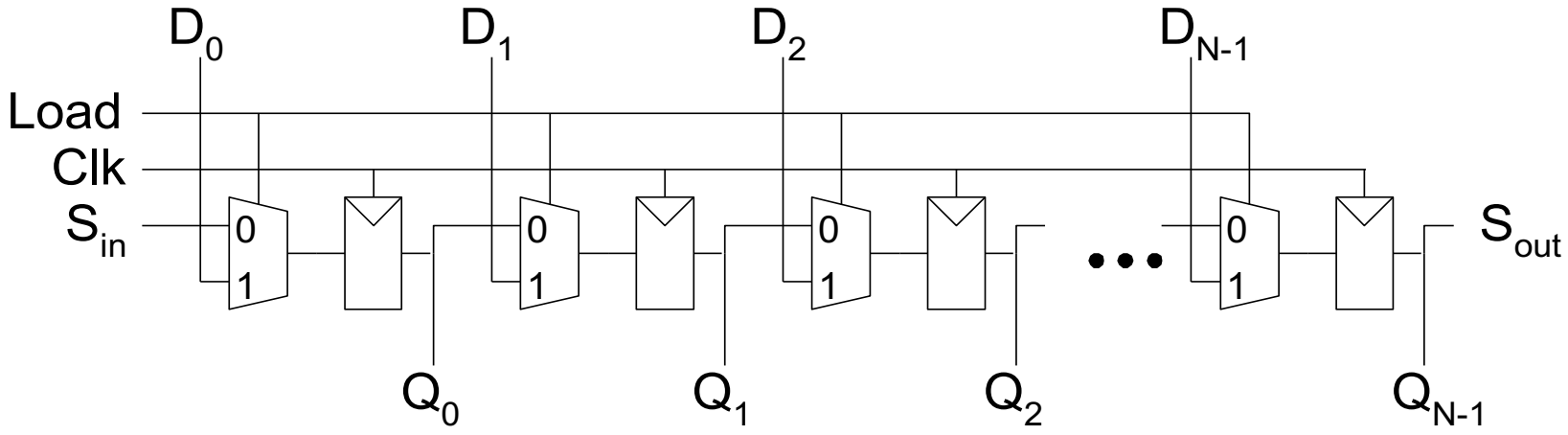
- Shift a new bit in on each clock edge
- Shift a bit out on each clock edge
- Serial-to-parallel converter: converts serial input (S_{in}) to parallel output ($Q_{0:N-1}$)





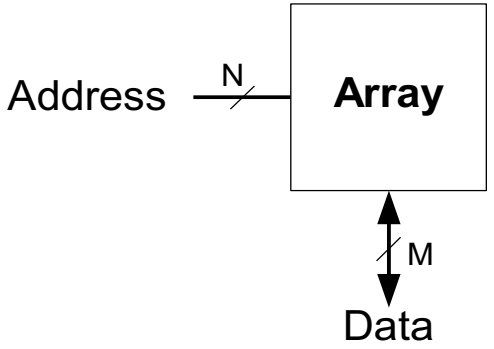
Shift Register with Parallel Load

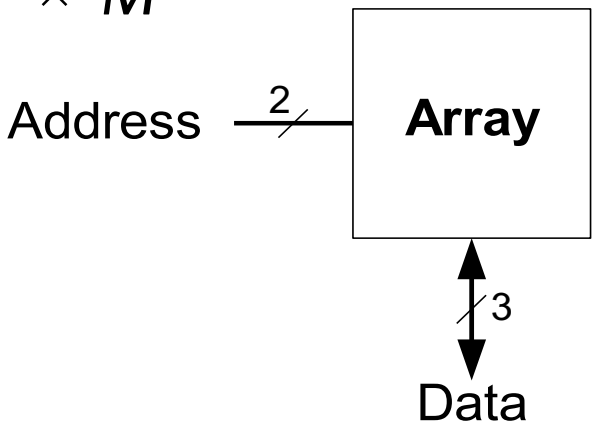
- When *Load* = 1, acts as a normal *N*-bit register
- When *Load* = 0, acts as a shift register
- Now can act as a *serial-to-parallel converter* (S_{in} to $Q_{0:N-1}$) or a *parallel-to-serial converter* ($D_{0:N-1}$ to S_{out})

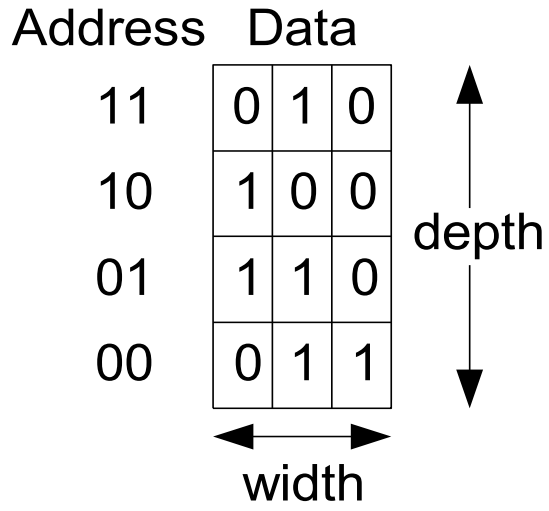


Memory Arrays

- 2-dimensional array of bit cells
- Each bit cell stores one bit
- *N* address bits and *M* data bits
  - 2^N rows and *M* columns
  - **Depth:** number of rows (number of words)
  - **Width:** number of columns (size of word)
  - **Array size:** depth × width = $2^N \times M$







Types of Memory

- Random access memory (RAM): **volatile**
- Read only memory (ROM): **nonvolatile**

RAM: Random Access Memory

- **Volatile:** loses its data when power off
- Read and written quickly
- Main memory in your computer is RAM (DRAM)

Historically called *random* access memory because any data word accessed as easily as any other (in contrast to sequential access memories such as a tape recorder)

ROM: Read Only Memory

- **Nonvolatile:** retains data when power off
- Read quickly, but writing is impossible or slow
- Flash memory in cameras, thumb drives, and digital cameras are all ROMs

Historically called *read only* memory because ROMs were written at manufacturing time or by burning fuses. Once ROM was configured, it could not be written again. This is no longer the case for Flash memory and other types of ROMs.

Types of RAM

- **DRAM** (Dynamic random access memory)
- **SRAM** (Static random access memory)
- Differ in how they store data:
  - DRAM uses a capacitor
  - SRAM uses cross-coupled inverters

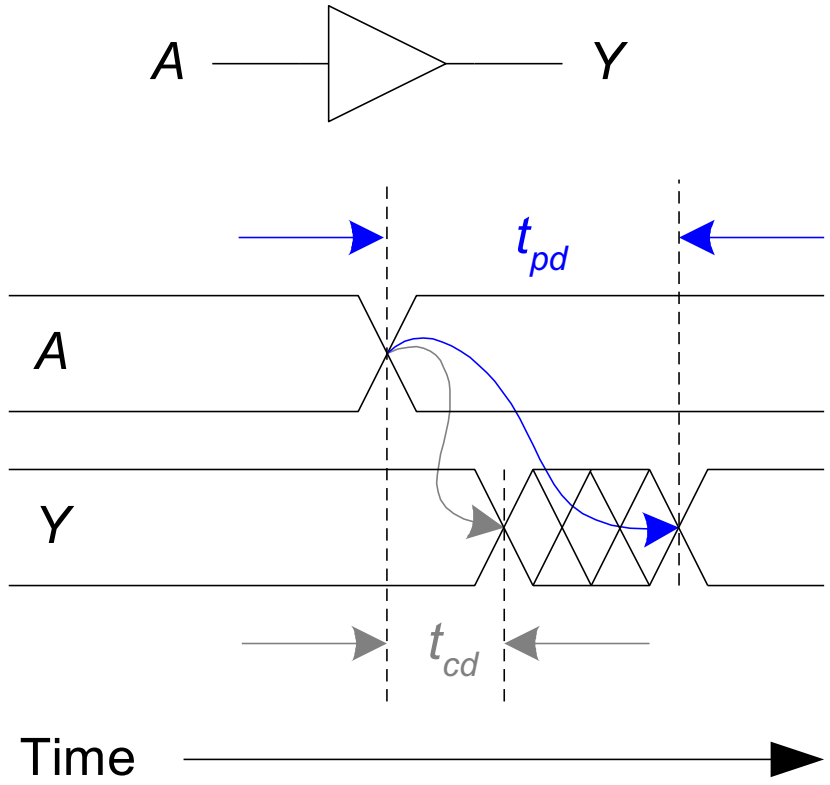
Types of ROMs

Type	Name	Description
ROM	Read Only Memory	Chip is hardwired with presence or absence of transistors. Changing requires building a new chip.
PROM	Programmable ROM	Fuses in series with each transistor are blown to program bits. Can't be changed after programming.
EPROM	Electrically Programmable ROM	Charge is stored on a floating gate to activate or deactivate transistor. Erasing requires exposure to UV light.
EEPROM	Electrically Erasable Programmable ROM	Like EPROM, but erasing can be done electrically.
Flash	Flash Memory	Like EEPROM, but erasing is done on large blocks to amortize cost of erase circuit. Low cost per bit, dominates nonvolatile storage today.

Synchronous Digital Design Review

Timing Review

- **Propagation delay:** t_{pd} = max delay from input to output
- **Contamination delay:** t_{cd} = min delay from input to output

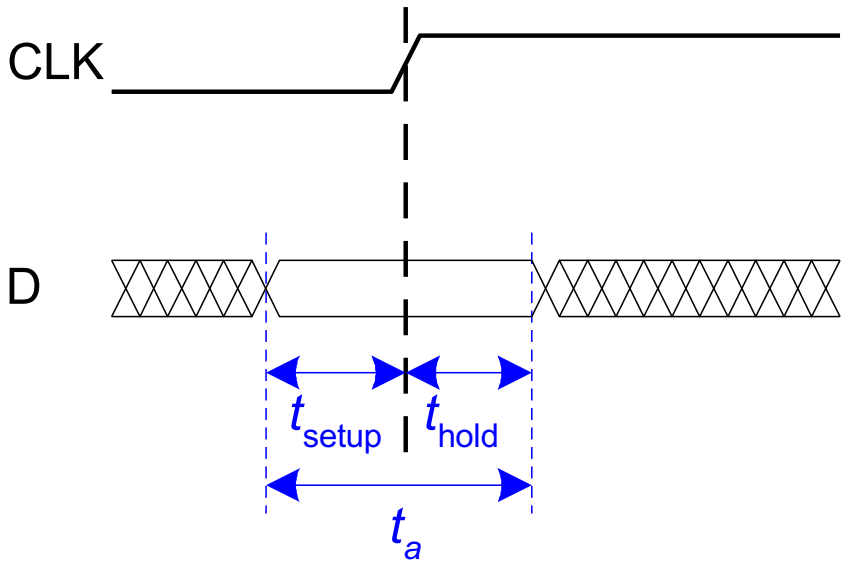


Flip-flop sampling

- Flip-flop samples *D* at clock edge
- *D* must be stable when sampled
- Similar to a photograph, *D* must be stable around clock edge
- If not, metastability can occur

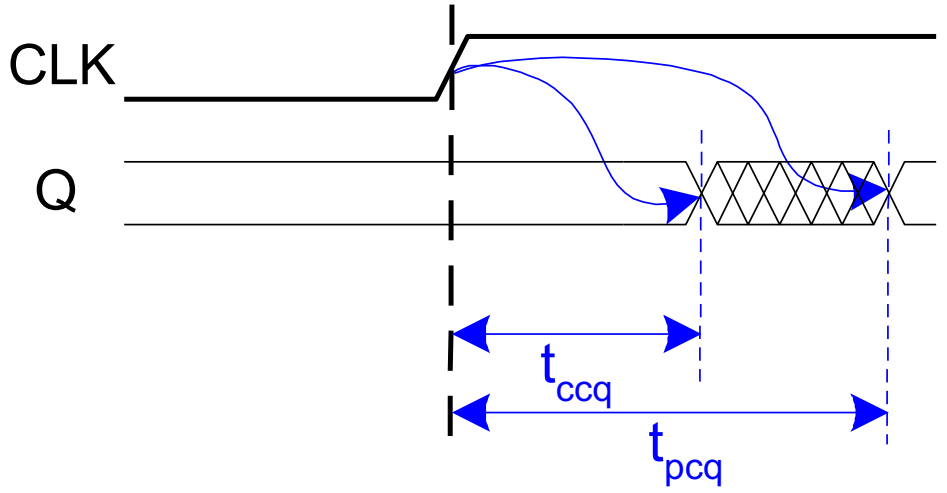
Input Timing Constraints

- **Setup time:** t_{setup} = time *before* clock edge data must be stable (i.e. not changing)
- **Hold time:** t_{hold} = time *after* clock edge data must be stable
- **Aperture time:** t_a = time *around* clock edge data must be stable (t_a = t_{setup} + t_{hold})



Output Timing Parameters

- **Propagation delay: t_{pcq}** = time after clock edge that the output *Q* is guaranteed to be stable (i.e., to stop changing)
- **Contamination delay: t_{ccq}** = time after clock edge that *Q* might be unstable (i.e., start changing)

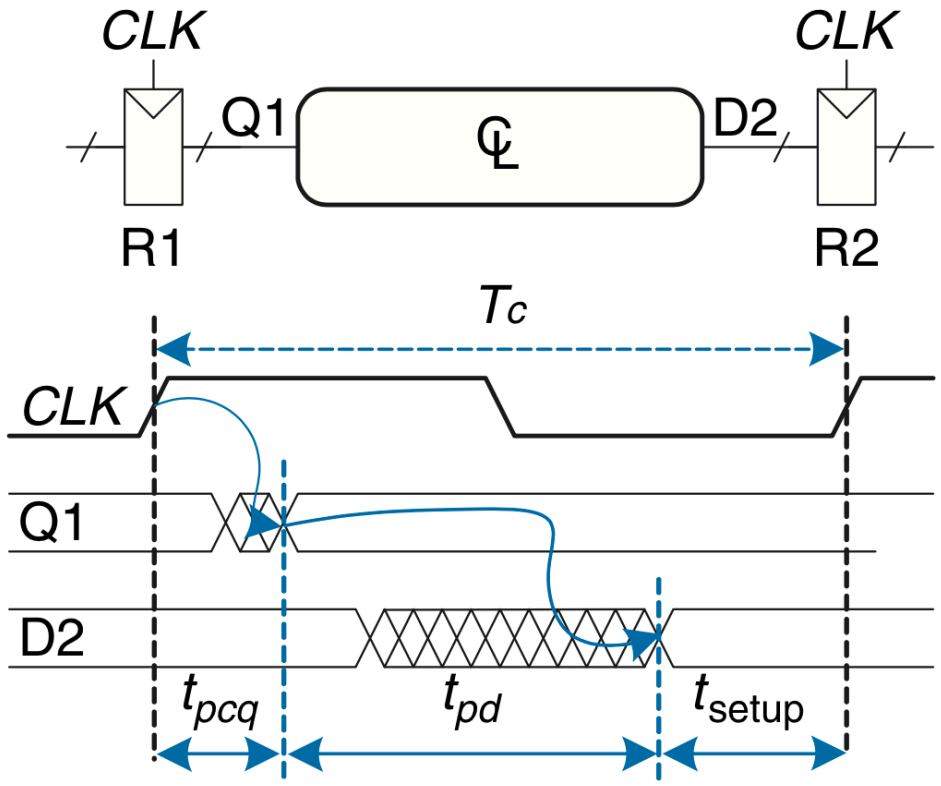


Dynamic Discipline

- Synchronous sequential circuit inputs must be stable during aperture (setup and hold) time around clock edge
- Specifically, inputs must be stable
  - at least t_{setup} before the clock edge
  - at least until t_{hold} after the clock edge

Dynamic Discipline

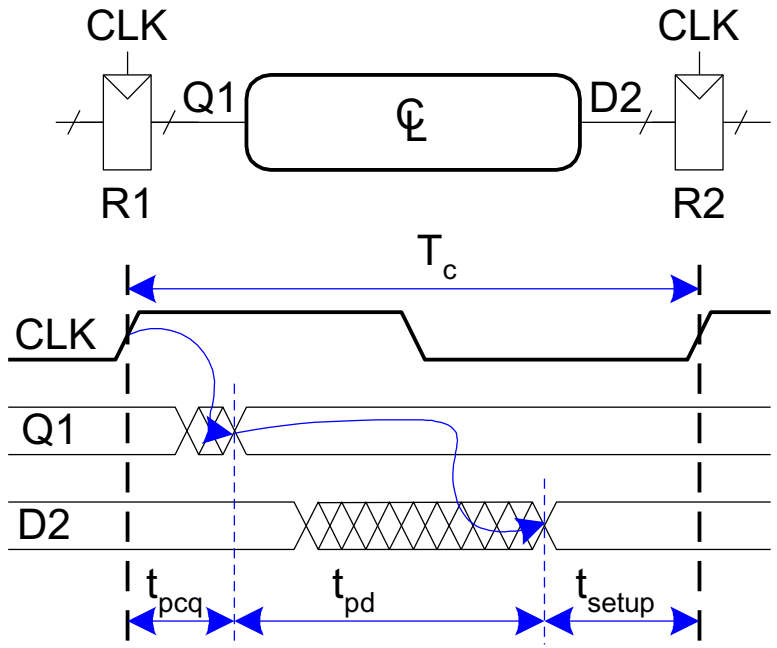
- The delay between registers has a **minimum** and **maximum** delay, dependent on the delays of the circuit elements



DDCA p. 143

Setup Time Constraint

- Depends on the **maximum** delay from register R1 through combinational logic to R2
- The input to register R2 must be stable at least t_{setup} before clock edge



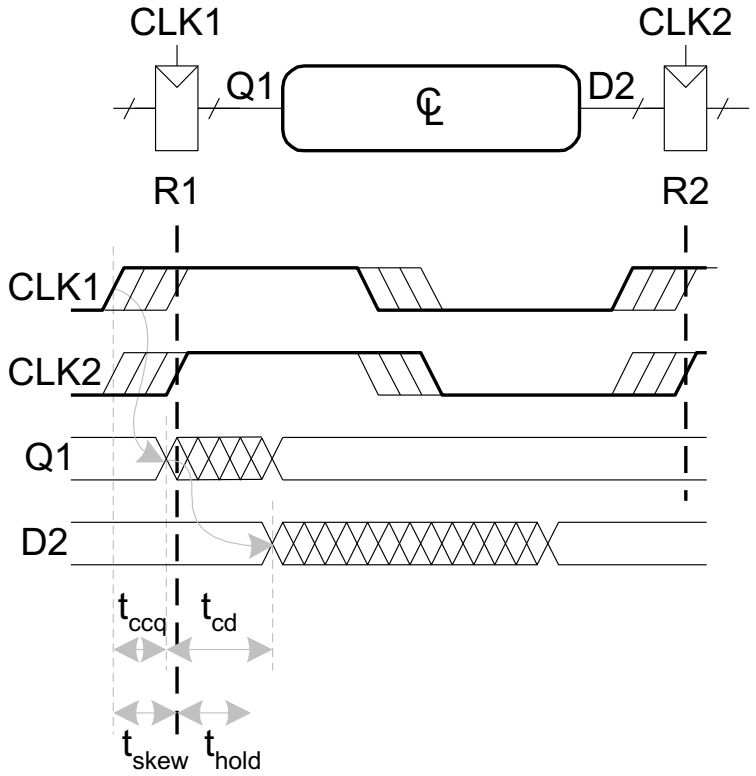
$$T_c \geq t_{pcq} + t_{pd} + t_{setup}$$

$$t_{pd} \leq T_c - (t_{pcq} + t_{setup})$$

$(t_{pcq} + t_{setup})$: *sequencing overhead*

Hold Time Constraint

- Depends on the **minimum** delay from register R1 through the combinational logic to R2
- The input to register R2 must be stable for at least t_{hold} after the clock edge



$$t_{ccq} + t_{cd} > t_{hold} + t_{skew}$$

$$t_{cd} > t_{hold} + t_{skew} - t_{ccq}$$

Timing Constraints

- These are important to consider when you are dealing with digital devices, especially when interfacing several devices.
- Are your devices operating on different clocks?
- How fast can you run your MCU?

How can we configure the clocks for 84 MHz?

- Turn on HSE (8 MHz source routed from ST-LINK) and HSEBYP (in RCC_CR)
- Configure PLL (RCC_PLLCFGR)
  - M = 8 (VCO_{in} = HSE/8 = 8 MHz/8 = 1 MHz)
  - N = 336 (VCO_{out} = VCO_{in} * 336 = 1 MHz * 336 = 336 MHz)
  - P = 4 (PLLCLK = VCO_{out}/4 = 84 MHz)
- Set the PLL as the clock source (RCC_CFGR)
- Try this out in your demo code

Flash speed

VDD = 3.3 V provided from regulator on Nucleo board

Table 15. Features depending on the operating power supply range

Operating power supply range	ADC operation	Maximum Flash memory access frequency with no wait states ($f_{Flashmax}$)	Maximum Flash memory access frequency with wait states (1)(2)	I/O operation	Clock output frequency on I/O pins(3)	Possible Flash memory operations
V_{DD} =1.7 to 2.1 V(4)	Conversion time up to 1.2 Msps	20 MHz(5)	84 MHz with 4 wait states	– No I/O compensation	up to 30 MHz	8-bit erase and program operations only
V_{DD} = 2.1 to 2.4 V	Conversion time up to 1.2 Msps	22 MHz	84 MHz with 3 wait states	– No I/O compensation	up to 30 MHz	16-bit erase and program operations
V_{DD} = 2.4 to 2.7 V	Conversion time up to 2.4 Msps	24 MHz	84 MHz with 3 wait states	– I/O compensation works	up to 48 MHz	16-bit erase and program operations
V_{DD} = 2.7 to 3.6 V(6)	Conversion time up to 2.4 Msps	30 MHz	84 MHz with 2 wait states	– I/O compensation works	– up to 84 MHz when V_{DD} = 3.0 to 3.6 V – up to 48 MHz when V_{DD} = 2.7 to 3.0 V	32-bit erase and program operations

STM32F401RE Datasheet p. 61

Flash speed

3.4 Read interface

3.4.1 Relation between CPU clock frequency and Flash memory read time

To correctly read data from Flash memory, the number of wait states (LATENCY) must be correctly programmed in the Flash access control register (FLASH_ACR) according to the frequency of the CPU clock (HCLK) and the supply voltage of the device.

Table 6. Number of wait states according to CPU clock (HCLK) frequency

Wait states (WS) (LATENCY)	HCLK (MHz)			
	Voltage range 2.7 V - 3.6 V	Voltage range 2.4 V - 2.7 V	Voltage range 2.1 V - 2.4 V	Voltage range 1.71 V - 2.1 V
0 WS (1 CPU cycle)	0 < HCLK≤ 30	0 < HCLK ≤ 24	0 < HCLK ≤ 18	0 < HCLK ≤ 16
1 WS (2 CPU cycles)	30 < HCLK ≤ 60	24 < HCLK ≤ 48	18 < HCLK ≤ 36	16 <HCLK ≤ 32
2 WS (3 CPU cycles)	60 < HCLK ≤ 84	48 < HCLK ≤ 72	36 < HCLK ≤ 54	32 < HCLK ≤ 48
3 WS (4 CPU cycles)		72 < HCLK ≤ 84	54 < HCLK ≤ 72	48 < HCLK ≤ 64
4 WS (5 CPU cycles)	-	-	72 < HCLK ≤ 84	64 < HCLK ≤ 80
5 WS (6 CPU cycles)	-	-	-	80 < HCLK ≤ 84

After reset, the CPU clock frequency is 16 MHz and 0 wait state (WS) is configured in the FLASH_ACR register.

It is highly recommended to use the following software sequences to tune the number of wait states needed to access the Flash memory with the CPU frequency.

STM32F401RE Reference Manual p. 46

Setting flash latency

3.8.1 Flash access control register (FLASH_ACR)

The Flash access control register is used to enable/disable the acceleration features and control the Flash memory access time according to CPU frequency.

Address offset: 0x00
Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DCRST	ICRST	DCEN	ICEN	PRFTEN	Reserved				LATENCY			
			rw	w	rw	rw	rw					rw	rw	rw	rw

Bits 3:0 **LATENCY:** Latency

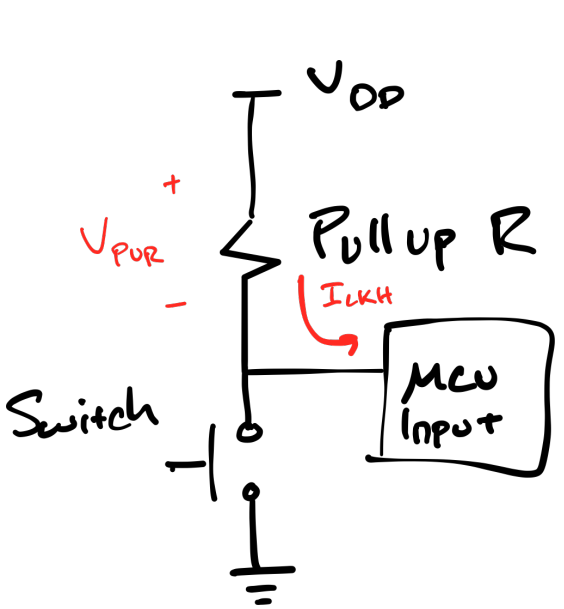
These bits represent the ratio of the CPU clock period to the Flash memory access time.
0000: Zero wait state
0001: One wait state
0010: Two wait states
-
-
-
1110: Fourteen wait states
1111: Fifteen wait states

STM32F401RE Reference Manual p. 60

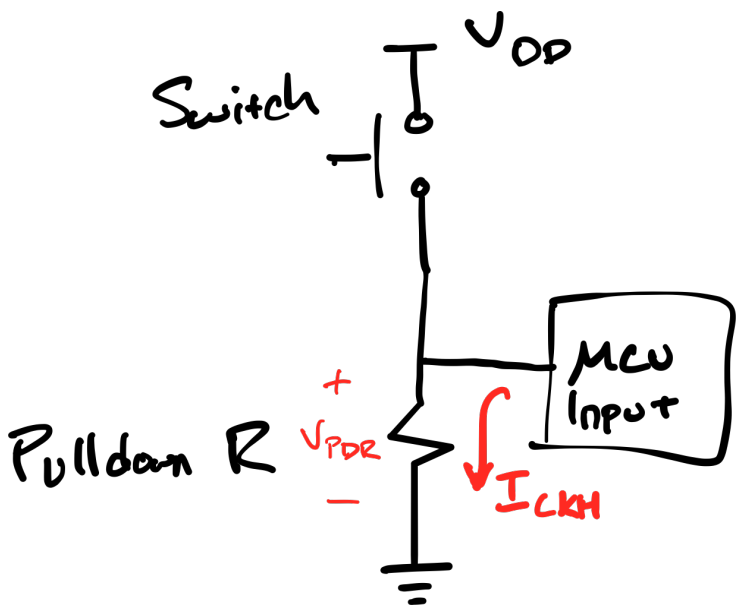
Pull Up/Pull Down Resistors

- Prevents input from floating – pulls value to either logic high (pull up) or logic low (pull down)
- Selection rationale
  - Can't be too small or draws a significant current and power
  - Can't be too big, or may generate too much voltage from leakage current and generate invalid logic levels.
- Typically around 10-50 kΩ

Pull up



Pull down



I/O static characteristics

Table 54. I/O static characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IL}	FT, and NRST I/O input low level voltage	1.7 V≤ V_{DD}≤ 3.6 V	-	-	0.35V_{DD}–0.04(1) 0.3V_{DD}(2)	V
	BOOT0 I/O input low level voltage	1.75 V≤ V_{DD} ≤ 3.6 V, -40 °C≤ T_A ≤ 105 °C	-	-	0.1V_{DD}+0.1	
		1.7 V≤ V_{DD} ≤ 3.6 V, 0 °C≤ T_A ≤ 105 °C	-	-		
V_{IH}	FT and NRST I/O input high level voltage(5)	1.7 V≤ V_{DD}≤ 3.6 V	0.45V_{DD}+0.3(1) 0.4V_{DD}(2)	-	-	V
	BOOT0 I/O input high level voltage	1.75 V≤ V_{DD} ≤ 3.6 V, -40 °C≤ T_A ≤ 105 °C	0.17V_{DD}+0.7(1)	-	-	
		1.7 V≤ V_{DD} ≤ 3.6 V, 0 °C≤ T_A ≤ 105 °C				

$$R_{max-PU} = \frac{V_{DD} - V_{IH}}{I_{IH}} = \frac{3.3\ V - 1.785\ V}{1\times10^{-6}\ A} \approx 1.5\ \text{M}\Omega$$

$$R_{max-PD} = \frac{V_{IL} - 0}{I_{IH}} = \frac{1\ V - 0\ V}{1\times10^{-6}\ A} \approx 1\ \text{M}\Omega$$

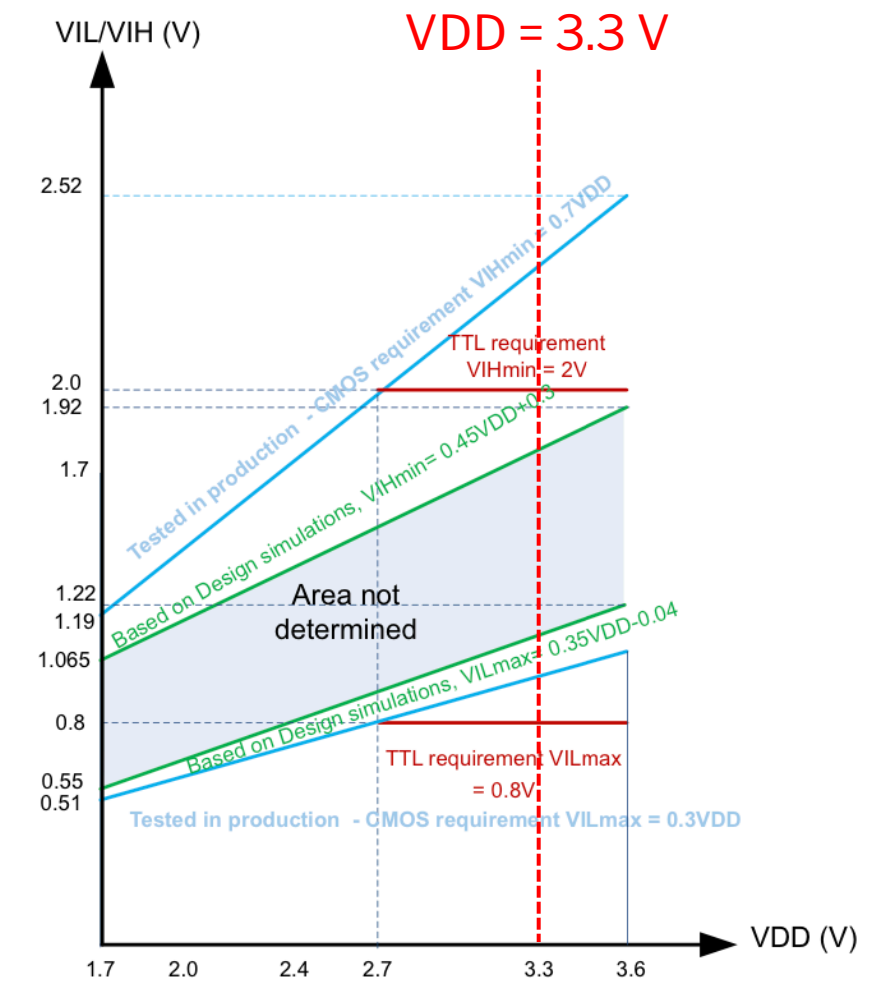
Table 54. I/O static characteristics (continued)

Symbol	Parameter		Conditions	Min	Typ	Max	Unit
V_{HYS}	FT and NRST I/O input hysteresis		1.7 V≤ V_{DD}≤ 3.6 V	-	10% V_{DD}(3)	-	V
	BOOT0 I/O input hysteresis		1.75 V≤ V_{DD} ≤ 3.6 V, -40 °C≤ T_A ≤ 105 °C	-	100	-	mV
			1.7 V≤ V_{DD} ≤ 3.6 V, 0 °C≤ T_A ≤ 105 °C				
I_{lkg}	I/O input leakage current (4)		V_{SS} ≤ V_{IN} ≤ V_{DD}	-	-	±1	µA
	I/O FT input leakage current (5)		V_{IN} = 5 V	-	-	3	
R_{PU}	Weak pull-up equivalent resistor(6)	All pins except for PA10 (OTG_FS_ID)	V_{IN} = V_{SS}	30	40	50	kΩ
		PA10 (OTG_FS_ID)		7	10	14	
R_{PD}	Weak pull-down equivalent resistor(7)	All pins except for PA10 (OTG_FS_ID)	V_{IN} = V_{DD}	30	40	50	
		PA10 (OTG_FS_ID)		7	10	14	
C_{IO}(8)	I/O pin capacitance		-	-	5	-	pF

p. 91-92 STM32F401RE Datasheet

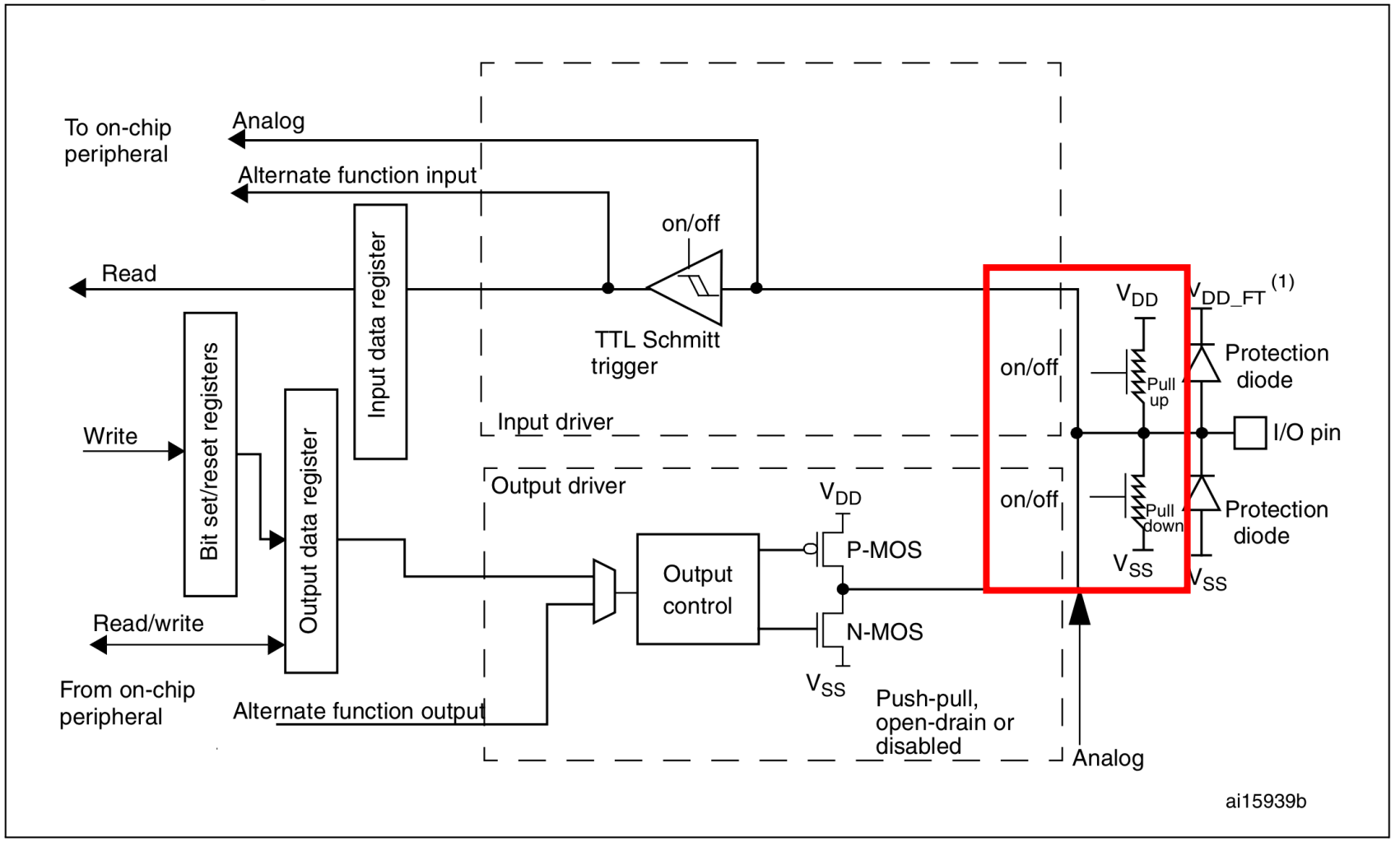
Figure with I/O input characteristics

Figure 30. FT I/O input characteristics



Pull up and pull down resistors in GPIO

Figure 16. Basic structure of a five-volt tolerant I/O port bit



Current Limiting Resistors

- When driving an external component like an LED, we want to make sure to not try to drive too much current from our I/O.

Max of 25 mA, so make sure to not draw more than this. LED output is bright around 10 mA, so for a 3.3 V output we can use:

$$R_{min} = \frac{3.3\ V - 0.7\ V}{10 \times 10^{-3}\ A} = 260\ \Omega$$

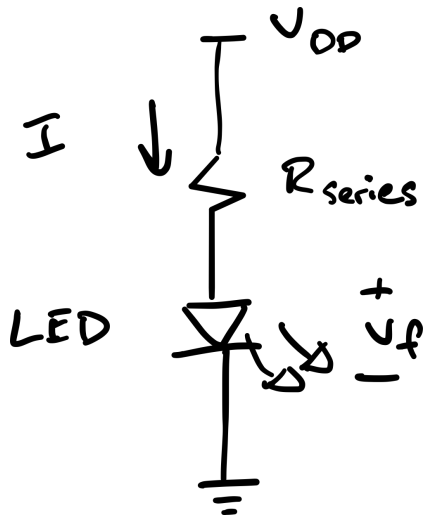


Table 12. Current characteristics

Symbol	Ratings	Max.	Unit
ΣI_{VDD}	Total current into sum of all V_{DD_x} power lines (source)(1)	160	mA
ΣI_{VSS}	Total current out of sum of all V_{SS_x} ground lines (sink)(1)	-160	
I_{VDD}	Maximum current into each V_{DD_x} power line (source)(1)	100	
I_{VSS}	Maximum current out of each V_{SS_x} ground line (sink)(1)	-100	
I_{IO}	Output current sunk by any I/O and control pin	25	
	Output current sourced by any I/O and control pin	-25	
ΣI_{IO}	Total output current sunk by sum of all I/O and control pins (2)	120	
	Total output current sourced by sum of all I/Os and control pins(2)	-120	
$I_{INJ(PIN)}$ (3)	Injected current on FT pins (4)	–5/+0	
	Injected current on NRST and B pins (4)		
$\Sigma I_{INJ(PIN)}$	Total injected current (sum of all I/O and control pins)(5)	±25	

Typically use something like a 330 Ω

Lab 3: Timers

Timers on STM32F401RE

- Up to 11 timers: up to six 16-bit, two 32-bit timers up to 84 MHz, each with up to four IC/OC/PWM or pulse counter and quadrature (incremental) encoder input, two watchdog timers (independent and window) and a SysTick timer

STM32F401RE Datasheet p. 1

TIM1 introduction

The advanced-control timers (TIM1) consist of a 16-bit auto-reload counter driven by a programmable prescaler.

TIM2 to TIM5 introduction

The general-purpose timers consist of a 16-bit or 32-bit auto-reload counter driven by a programmable prescaler.

They may be used for a variety of purposes, including measuring the pulse lengths of input signals (*input capture*) or generating output waveforms (*output compare and PWM*).

Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the RCC clock controller prescalers.

The timers are completely independent, and do not share any resources. They can be synchronized together as described in *Section 13.3.15*.

TIM9/10/11 introduction

The TIM9/10/11 general-purpose timers consist of a 16-bit auto-reload counter driven by a programmable prescaler.

They may be used for a variety of purposes, including measuring the pulse lengths of input signals (input capture) or generating output waveforms (output compare, PWM).

Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the RCC clock controller prescalers.

The TIM9/10/11 timers are completely independent, and do not share any resources. They can be synchronized together as described in *Section 14.3.12*.

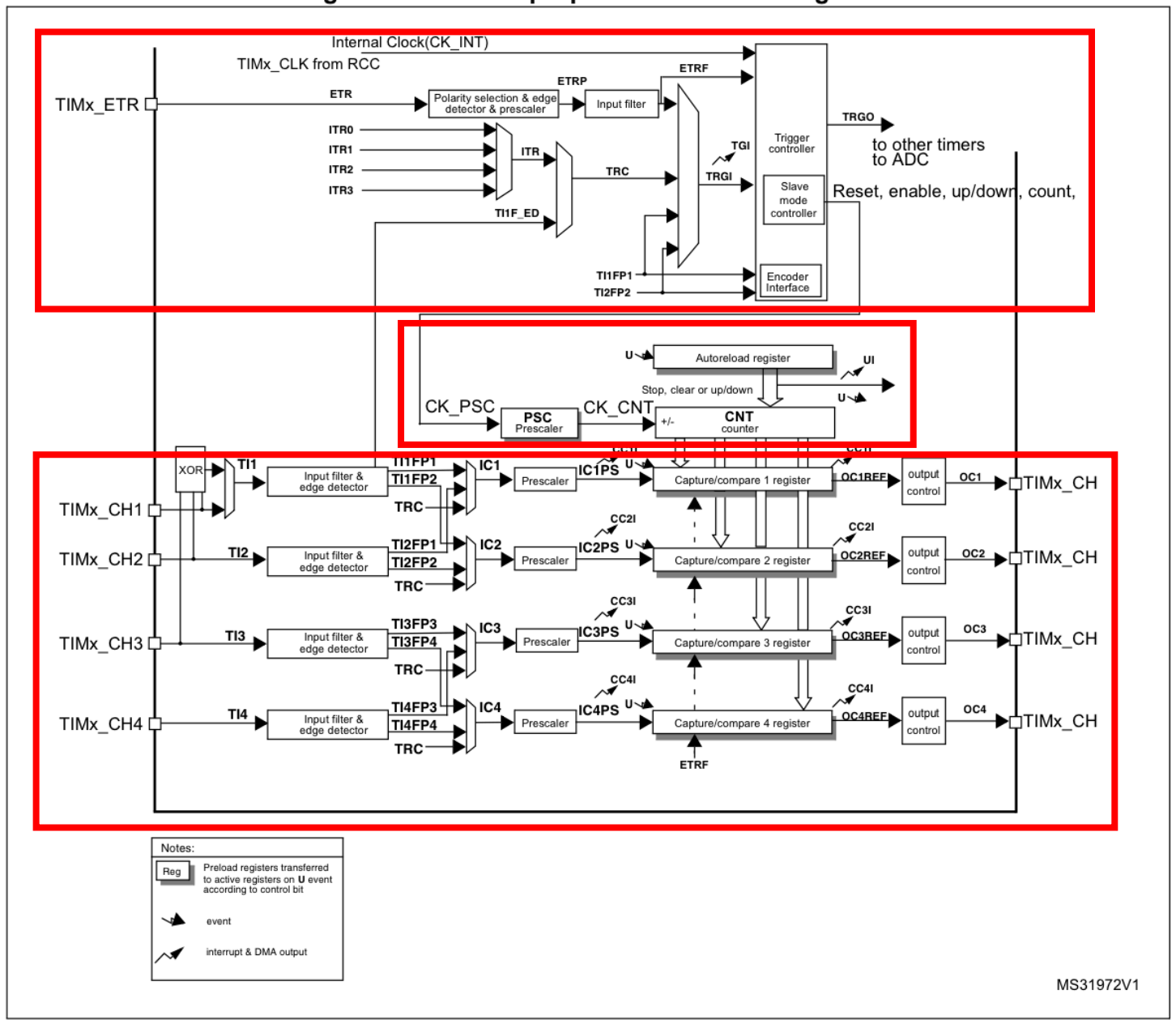
Block diagram

Clock Source and Trigger Controller

Time-base unit

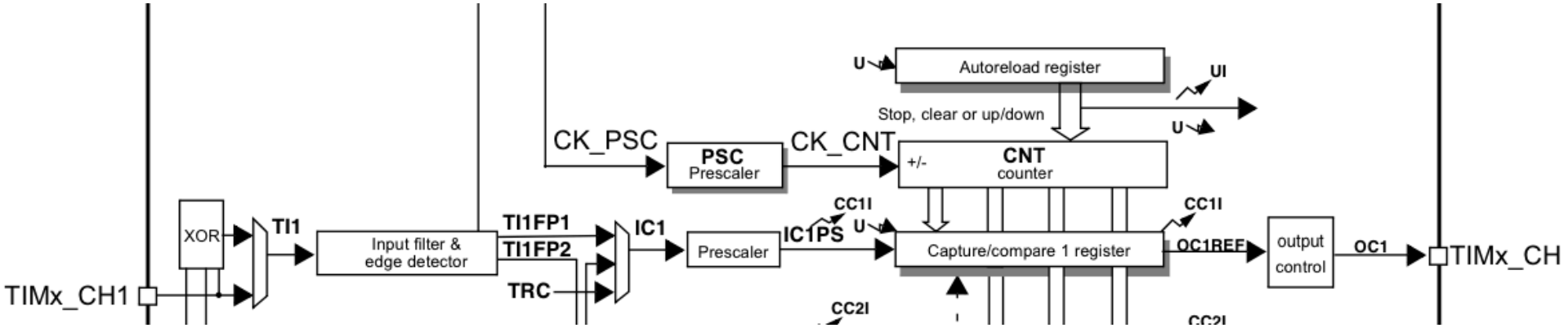
Counter Capture/Compare Channels

Figure 87. General-purpose timer block diagram



STM32F401RE Reference Manual p. 317

Single Timer Channel



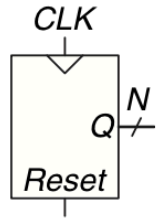


Figure 5.31 Counter symbol

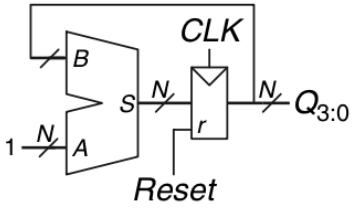


Figure 5.32 *N*-bit counter

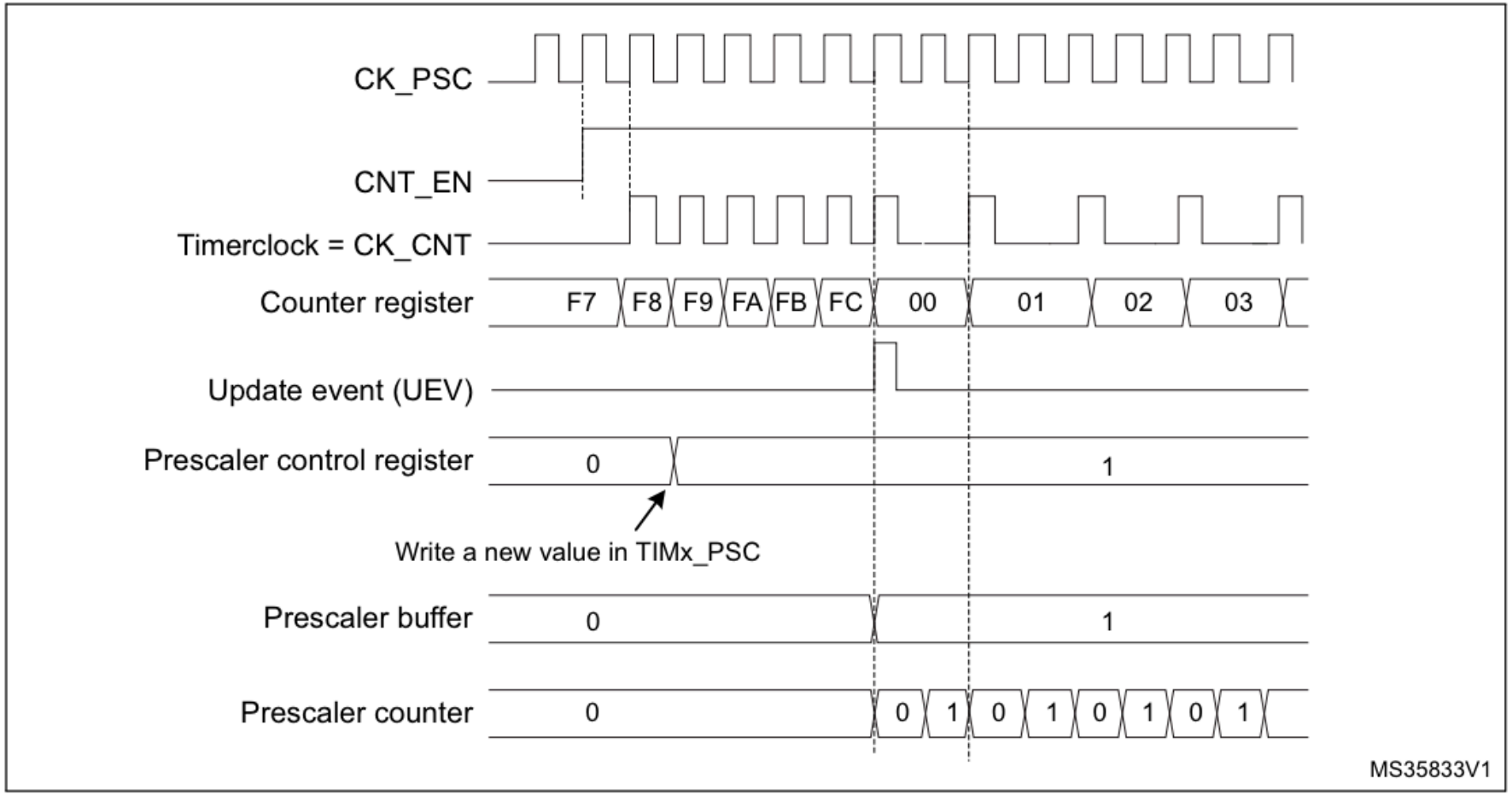
DDCA ARMed p. 260

Main blocks

- Prescaler Register (TIMx_PSC): counter to pre-scale the input clock signal (CK_PSC)
- Counter Register (TIMx_CNT): Counter which holds the current count value. Counts at the rate specified by the prescaled clock (CK_CNT)
- Auto-Reload Register (TIMx_ARR): Holds the max value for the counter
- Capture/compare register (CCR_X): Used to generate a signal that will be sent to the output control block for more advanced operations like pulse-width modulation (PWM)

Timer Example Diagrams: Prescaler

Figure 88. Counter timing diagram with prescaler division change from 1 to 2



Counter: Upcounting mode

TIMx_ARR=0x36

Figure 90. Counter timing diagram, internal clock divided by 1

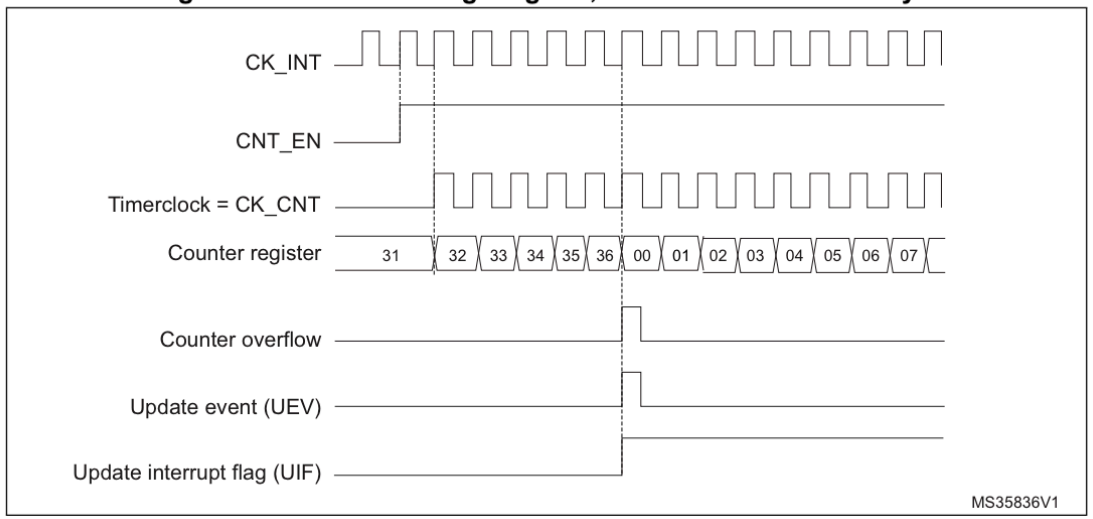
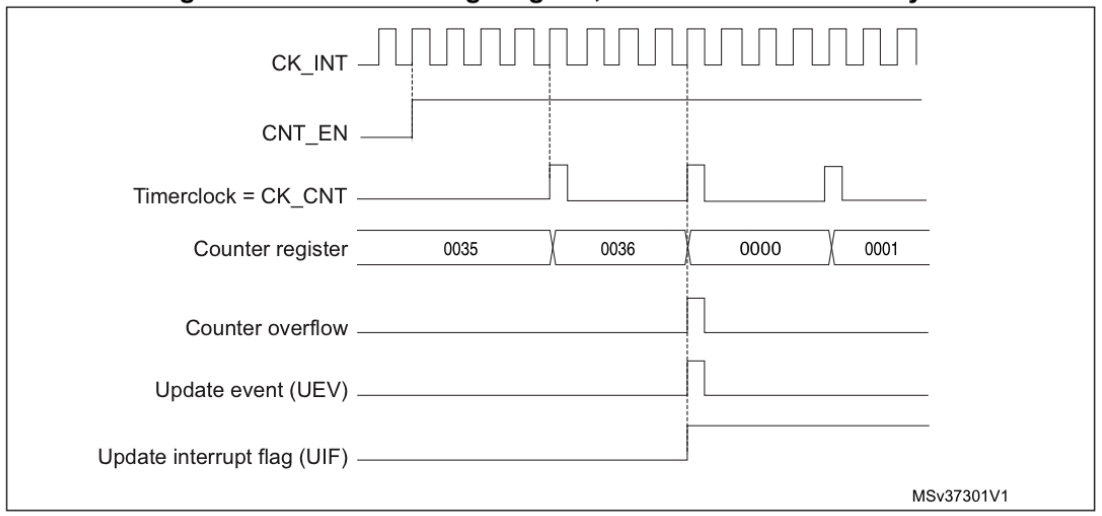


Figure 92. Counter timing diagram, internal clock divided by 4



Summary

- Recall timing and setup and hold time constraints when working with external digital circuitry
- Digital structures like shift registers are at the heart of serial interfaces (will discuss more on Wednesday!)
- Timers offer a way to precisely generate delays

Lecture Feedback

- What is the most important thing you learned in class today?
- What point was most unclear from lecture today?

https://forms.gle/Ay6MkpZ6x3xsW2Eb8