Serial Interfaces: Part 1

Lecture 8

Microprocessor-based Systems (E155)

Prof. Josh Brake

μP

Outline

- Serial Interfaces Overview
  - Advantages over parallel
  - Major considerations
  - Overview of protocols
- Serial Peripheral Interface
  - Description
  - STM32F401RE configuration
- MCP4801 DAC
  - Datasheet overview
- Lab 4 hints and suggestions

Learning Objectives

By the end of this lecture you will be able to

- List common specifications of a serial interface
- Articulate the tradeoffs between different serial protocols
- Explain the basic operation of SPI

Quiz

- Make a bulleted list of the general initialization procedure for a peripheral. What are the major steps?
- Serial interfaces often are idle high (the data or clock line is held at a non-zero voltage). Why do you think this is the case?
- How is it possible to read data in a serial interface without a shared clock (asynchronously)?
- Name a serial interface you've used recently.

https://docs.google.com/presentation/d/1ShVehgj6aX2Dr44j0UIh4JXsUJaZQ8HdsNg32T6HZcs/edit#slide=id.p

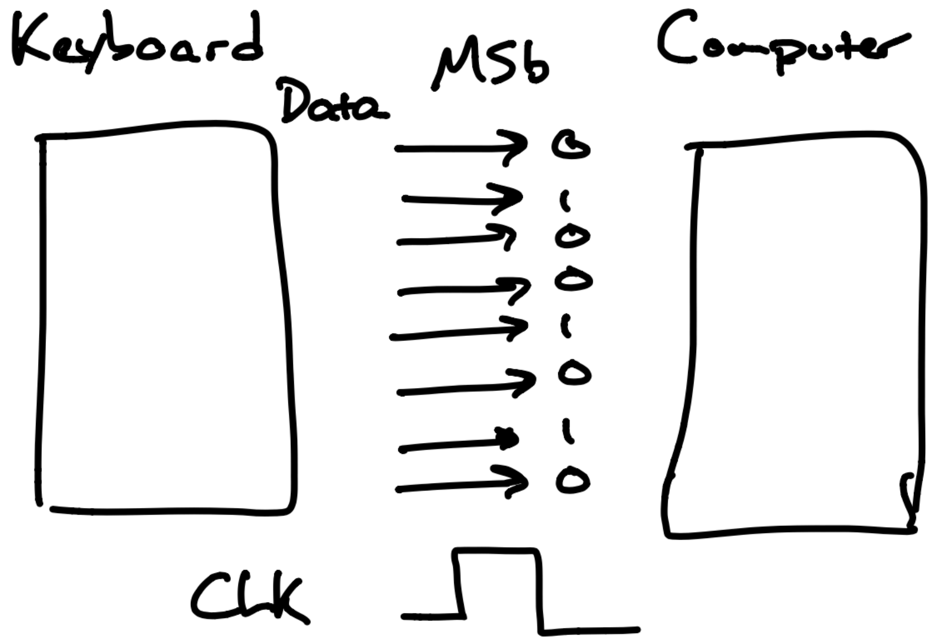
Serial Interfaces Overview

Motivation

- How can we interface a peripheral?

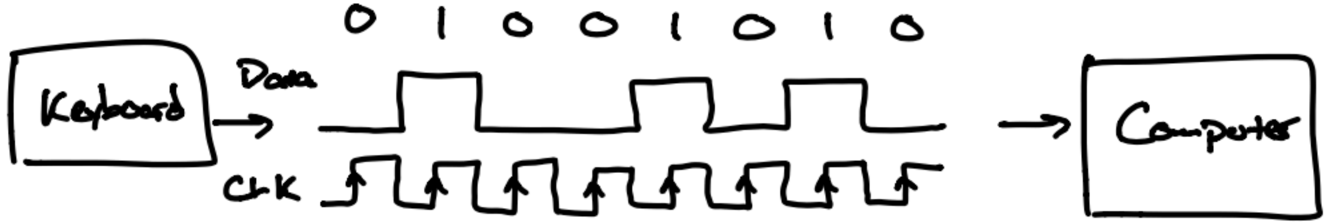
Imagine transmitting a character on a keyboard.

Capital J in ASCII is 74_{10} = 01001010_2



What if we repackage data in a stream?

- Essential multiplexing in time
- To send N bits, we only need 2 lines (CLK + Data) instead of 9
- Price we pay is time – but often worth it.



Example Serial Interfaces

- Universal Serial Bus (USB)
- Serial Peripheral Interface (SPI)
- Thunderbolt
- Universal Asynchronous Receiver Transmitter (UART)
- Inter-integrated circuit (I^2C)
- Ethernet

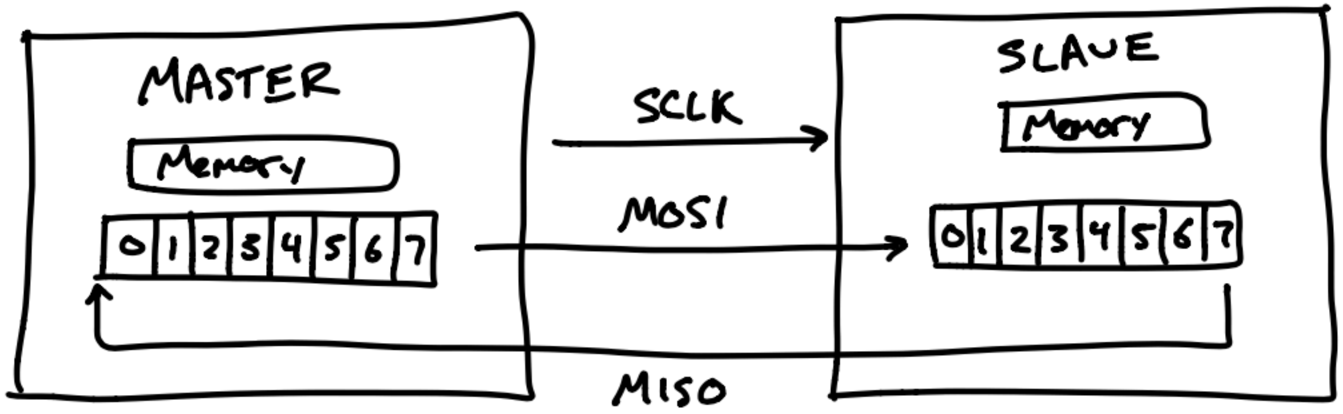
Basic Specifications

- Synchronous vs. Asynchronous – is there a shared clock signal?
- Number of wires
- Data transmission modes – simplex, full/half duplex
- Bandwith/speed/bitrate/baud rate
- Signal voltage levels (e.g., TTL, RS-232)
- Max cable length

Serial Peripheral Interface (SPI)

SPI Overview and History

- Developed in the mid-1980s by Motorola
- Used to interface with many peripherals like memory (SD cards, flash), displays, sensors (accelerometers, gyroscopes, temperature sensors, ADCs and DACs).
- Four-wire, synchronous serial bus

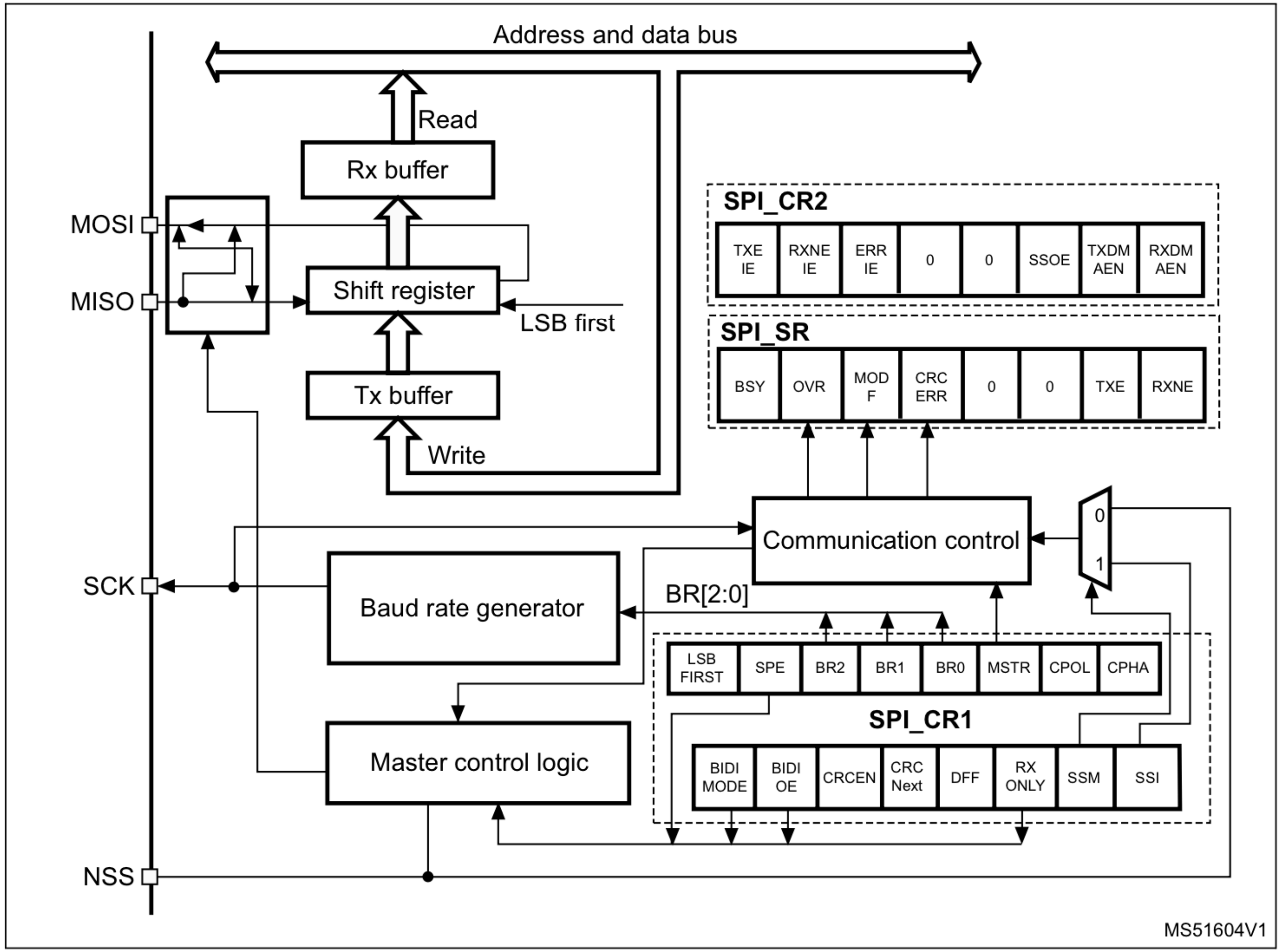


SCLK: Serial clock
MOSI: Master Out Slave In
MISO: Master In Slave Out
CE/CS/nCE/nCS: Chip select/enable

SPI Block Diagram

- MISO: Master In / Slave Out
- MOSI: Master Out / Slave In
- SCK: Serial Clock output
- NSS: Slave select (N means active low)

Figure 192. SPI block diagram



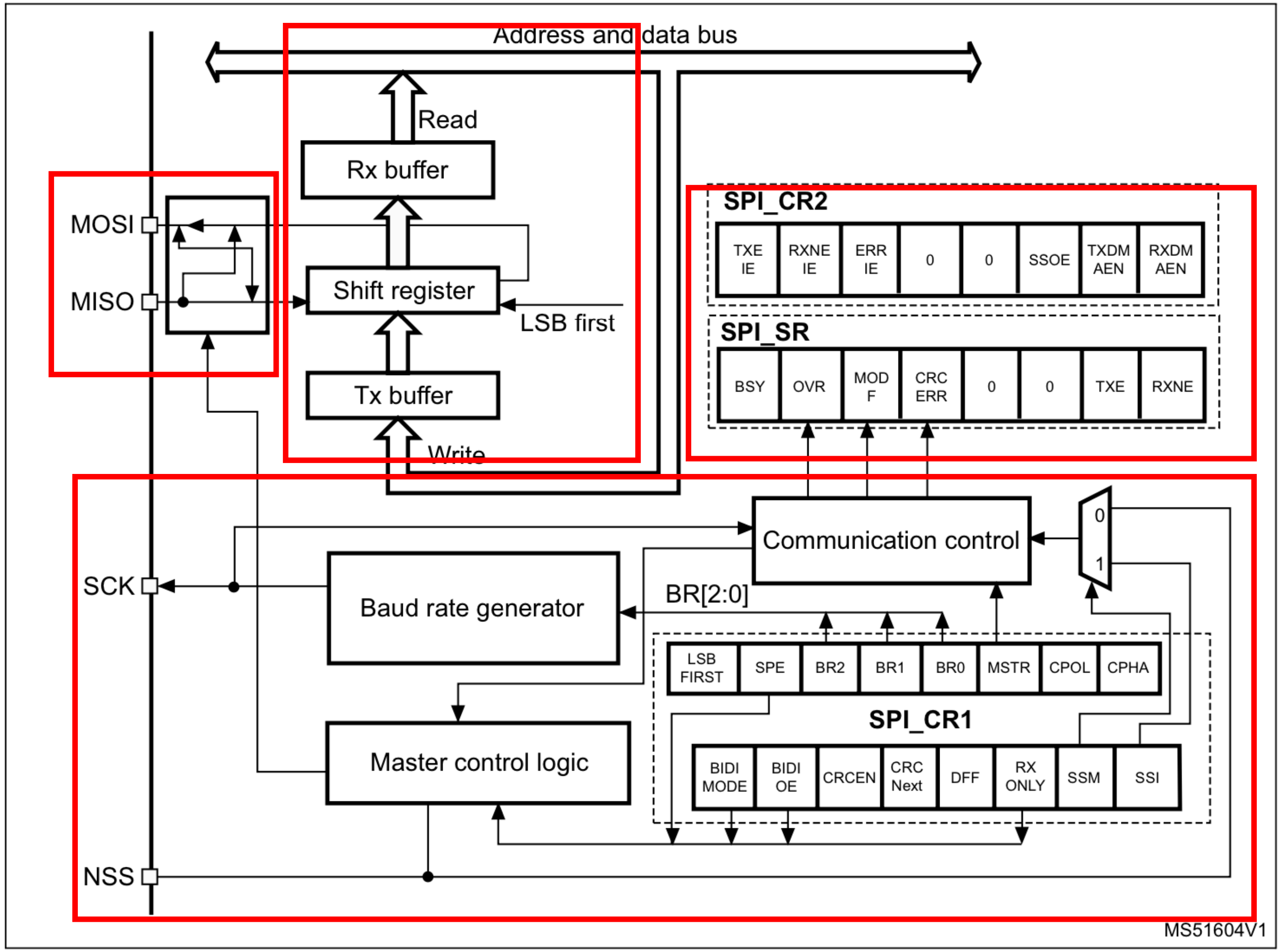
STM32F401RE Reference Manual p. 562

SPI Block Diagram in Detail

Data Registers and Buffers

Figure 192. SPI block diagram

Buffer Data Output



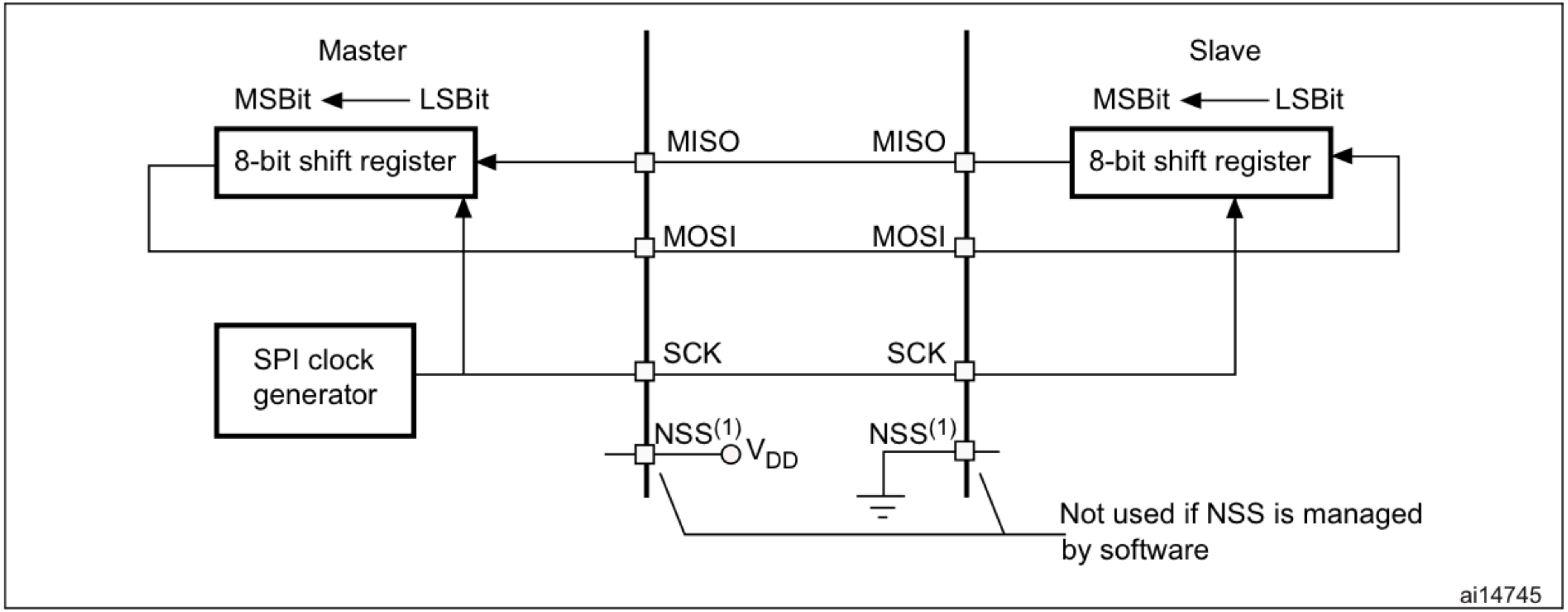
Control and Status Registers

Clocking and Synchronization

SPI on STM32

- Can be either master or slave (we will be using as master)

Figure 193. Single master/ single slave application

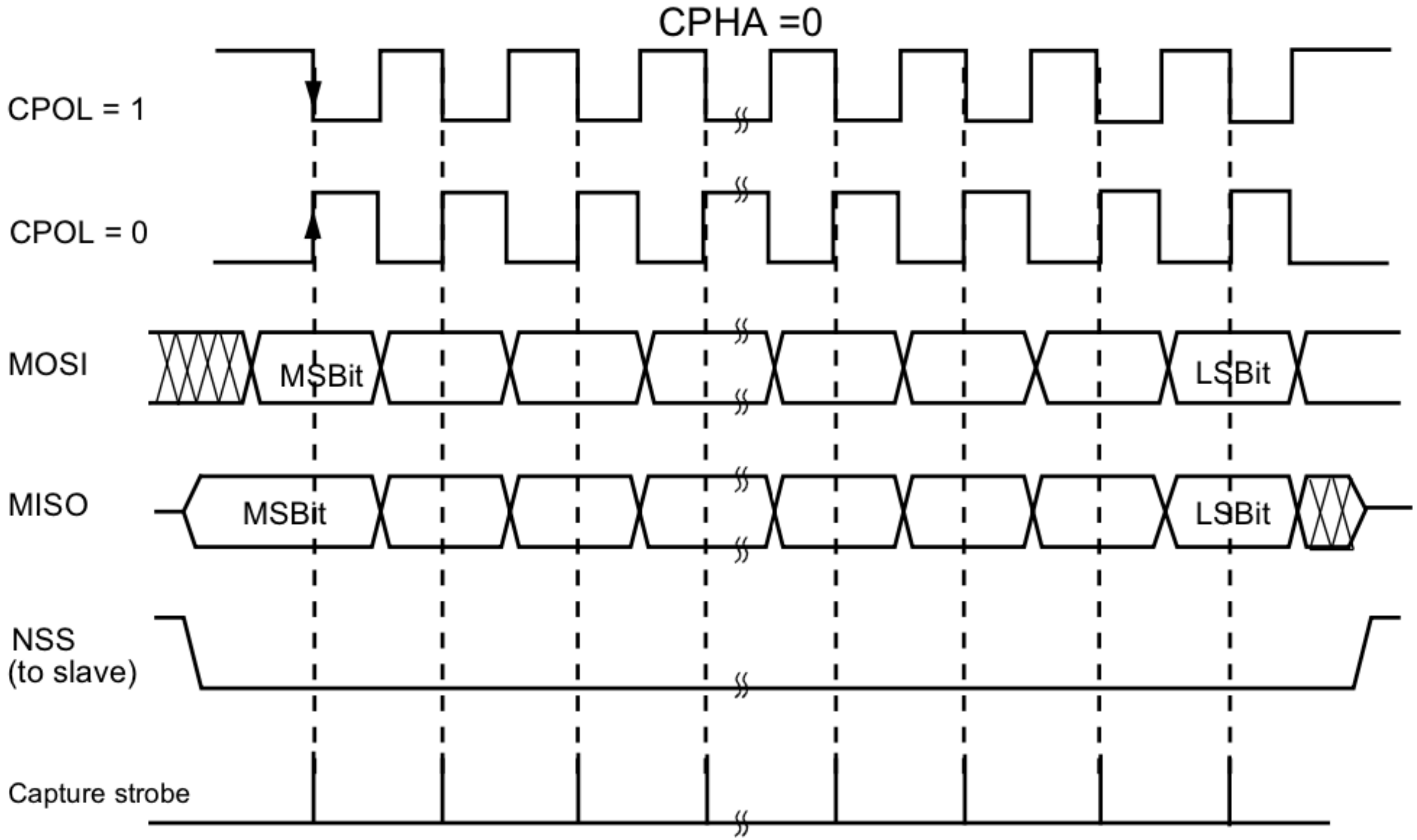


1. Here, the NSS pin is configured as an input.

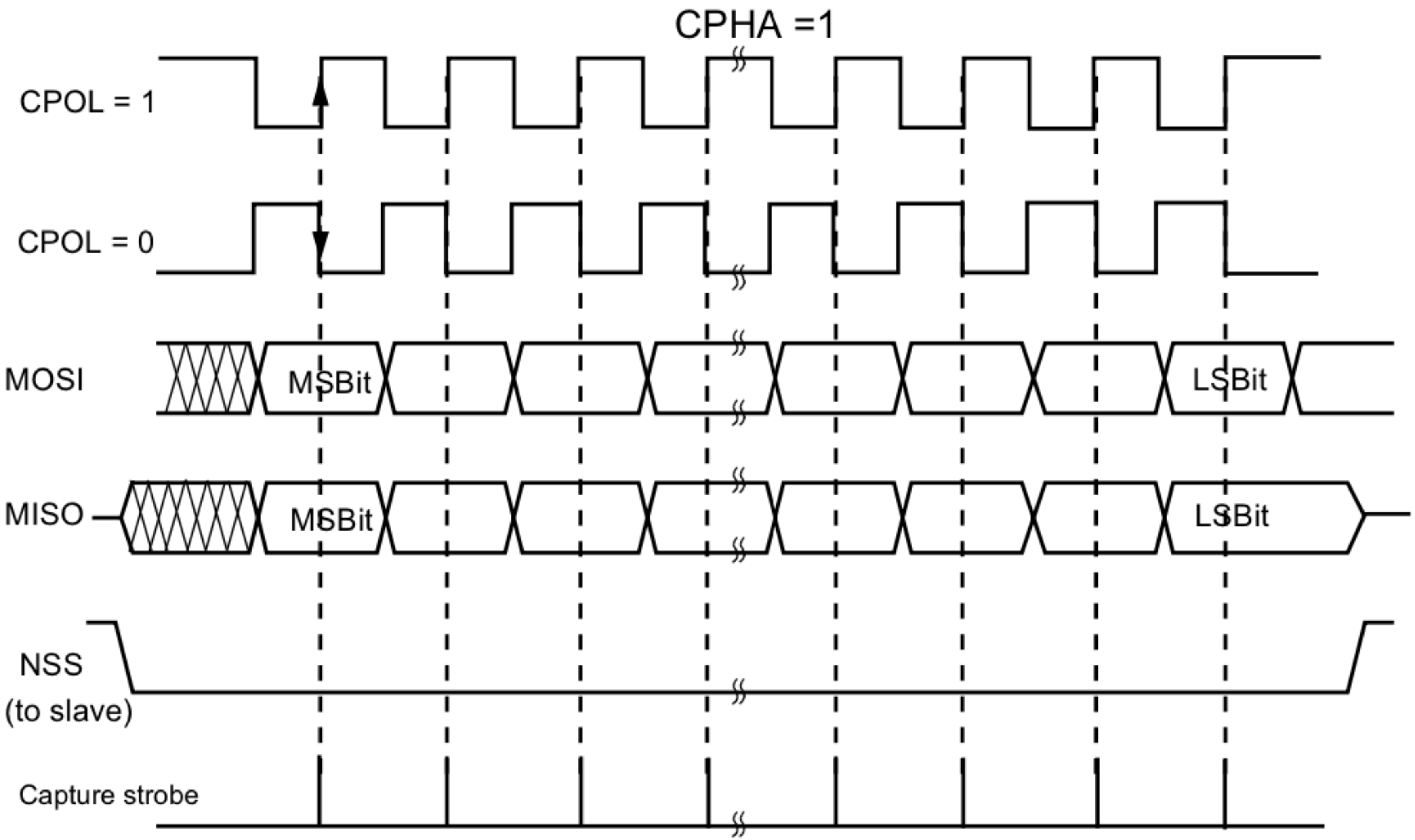
Clock Polarity and Phase

- Clock polarity (CPOL) refers to the state of the clock line at idle
  - 0: clock is low when idle
  - 1: clock is high when idle
- Clock phase (CPHA) refers to when data is sampled vs. when new data is shifted out
  - 0: the first clock transition is the first data capture edge
  - 1: the second clock transition is the first data capture edge
- The clock transition (rising or falling) depends on the clock polarity
- 4 combinations or modes (CPOL,CPHA) = (0,0), (0,1), (1,0), (1,1)
- Must pay attention to match this mode to the slave!

Clock Phase and Polarity: Clock Phase = 0



Clock Phase and Polarity: Clock Phase = 1



Basic Configuration in Master Mode

- Configure clock tree
- Turn on SPI clock domain
- Set SPI parameters
  - Clock rate using baud rate divisor
  - CPOL and CPHA to match slave
  - DFF to 8- or 16-bit data frame format
  - Set LSBFIRST bit to set whether lsb or msb is sent first (normally msb)
  - Configure the NSS pin (can either use software management or a separate GPIO set as an output and manually toggle it)
  - Set to master mode MSTR
- Enable SPI – Set SPE bit to 1

MCP4801 DAC and Lab 4

Interfacing with external chip: MCP4801 DAC

MICROCHIP **MCP4801/4811/4821**

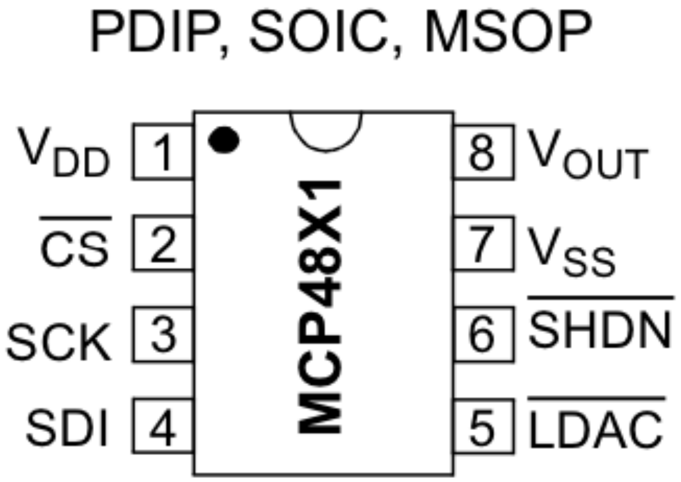
8/10/12-Bit Voltage Output Digital-to-Analog Converter with Internal V_{REF} and SPI Interface

22244B.pdf

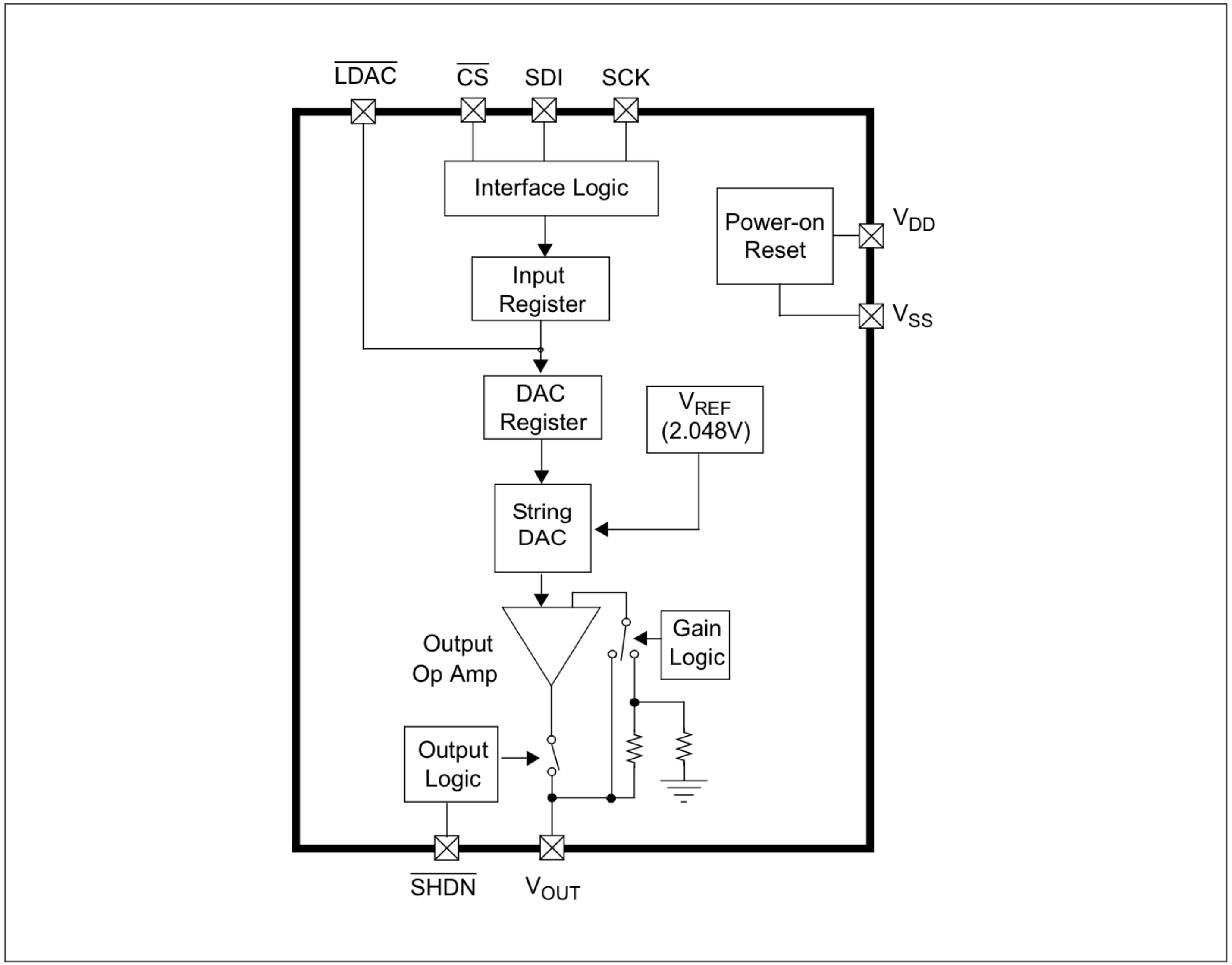
P/N	DAC Resolution	No. of Channel	Voltage Reference (V_{REF})
MCP4801	**8**	**1**	Internal (2.048V)
MCP4811	**10**	**1**	
MCP4821	**12**	**1**	
MCP4802	8	2	
MCP4812	10	2	
MCP4822	12	2	
MCP4901	8	1	External
MCP4911	10	1	
MCP4921	12	1	
MCP4902	8	2	
MCP4912	10	2	
MCP4922	12	2	

Note 1: The products listed here have similar AC/DC performances.

Block Diagram



Block Diagram



Electrical Characteristics

ELECTRICAL CHARACTERISTICS

Electrical Specifications: Unless otherwise indicated, V_{DD} = 5V, V_{SS} = 0V, V_{REF} = 2.048V, Output Buffer Gain (G) = 2x, RL = 5 kΩ to GND, C_L = 100 pF, T_A = -40 to +85°C. Typical values are at +25°C.

Parameters	Sym	Min	Typ	Max	Units	Conditions
Power Requirements						
Operating Voltage	V_{DD}	2.7	—	5.5		
Operating Current	I_{DD}	—	330	400	µA	All digital inputs are grounded, analog output (V_{OUT}) is unloaded. Code = 000h
Hardware Shutdown Current	I_{SHDN}	—	0.3	2	µA	POR circuit is turned off
Software Shutdown Current	I_{SHDN_SW}	—	3.3	6	µA	POR circuit remains turned on
Power-on Reset Threshold	V_{POR}	—	2.0	—	V	
DC Accuracy						
MCP4801						
Resolution	n	8	—	—	Bits	
INL Error	INL	-1	±0.125	1	LSb	
DNL	DNL	-0.5	±0.1	+0.5	LSb	**Note 1**

Absolute Maximum Ratings †

V_{DD} 6.5V

All inputs and outputs .... V_{SS} – 0.3V to V_{DD} + 0.3V

Current at Input Pins .... ±2 mA

Current at Supply Pins .... ±50 mA

Current at Output Pins .... ±25 mA

Storage temperature .... -65°C to +150°C

Ambient temp. with power applied .... -55°C to +125°C

ESD protection on all pins .... ≥ 4 kV (HBM), ≥ 400V (MM)

Maximum Junction Temperature (T_J) .... +150°C

Timing Specs

- Pay attention to max clock frequency.
- These specs are related to setup and hold time constraints!

AC CHARACTERISTICS (SPI TIMING SPECIFICATIONS)

Electrical Specifications: Unless otherwise indicated, V_{DD}= 2.7V – 5.5V, T_A= -40 to +125°C. Typical values are at +25°C.

Parameters	Sym	Min	Typ	Max	Units	Conditions
Schmitt Trigger High-Level Input Voltage (All digital input pins)	V_{IH}	0.7 V_{DD}	—	—	V	
Schmitt Trigger Low-Level Input Voltage (All digital input pins)	V_{IL}	—	—	0.2 V_{DD}	V	
Hysteresis of Schmitt Trigger Inputs	V_{HYS}	—	0.05 V_{DD}	—		
Input Leakage Current	$I_{LEAKAGE}$	-1	—	1	µA	$\overline{SHDN}$ = $\overline{LDAC}$ = $\overline{CS}$ = SDI = SCK = V_{DD} or V_{SS}
Digital Pin Capacitance (All inputs/outputs)	C_{IN}, C_{OUT}	—	10	—	pF	V_{DD} = 5.0V, T_A = +25°C, f_{CLK} = 1 MHz **(Note 1)**
Clock Frequency	F_{CLK}	—	—	20	MHz	T_A = +25°C **(Note 1)**
Clock High Time	t_{HI}	15	—	—	ns	**Note 1**
Clock Low Time	t_{LO}	15	—	—	ns	**Note 1**
$\overline{CS}$ Fall to First Rising CLK Edge	t_{CSSR}	40	—	—	ns	Applies only when $\overline{CS}$ falls with CLK high. **(Note 1)**
Data Input Setup Time	t_{SU}	15	—	—	ns	**Note 1**
Data Input Hold Time	t_{HD}	10	—	—	ns	**Note 1**
SCK Rise to $\overline{CS}$ Rise Hold Time	t_{CHS}	15	—	—	ns	**Note 1**
$\overline{CS}$ High Time	t_{CSH}	15	—	—	ns	**Note 1**
$\overline{LDAC}$ Pulse Width	t_{LD}	100	—	—	ns	**Note 1**
$\overline{LDAC}$ Setup Time	t_{LS}	40	—	—	ns	**Note 1**
SCK Idle Time before $\overline{CS}$ Fall	t_{IDLE}	40	—	—	ns	**Note 1**

Note 1: This parameter is ensured by design and not 100% tested.

Example Transmission Waveform

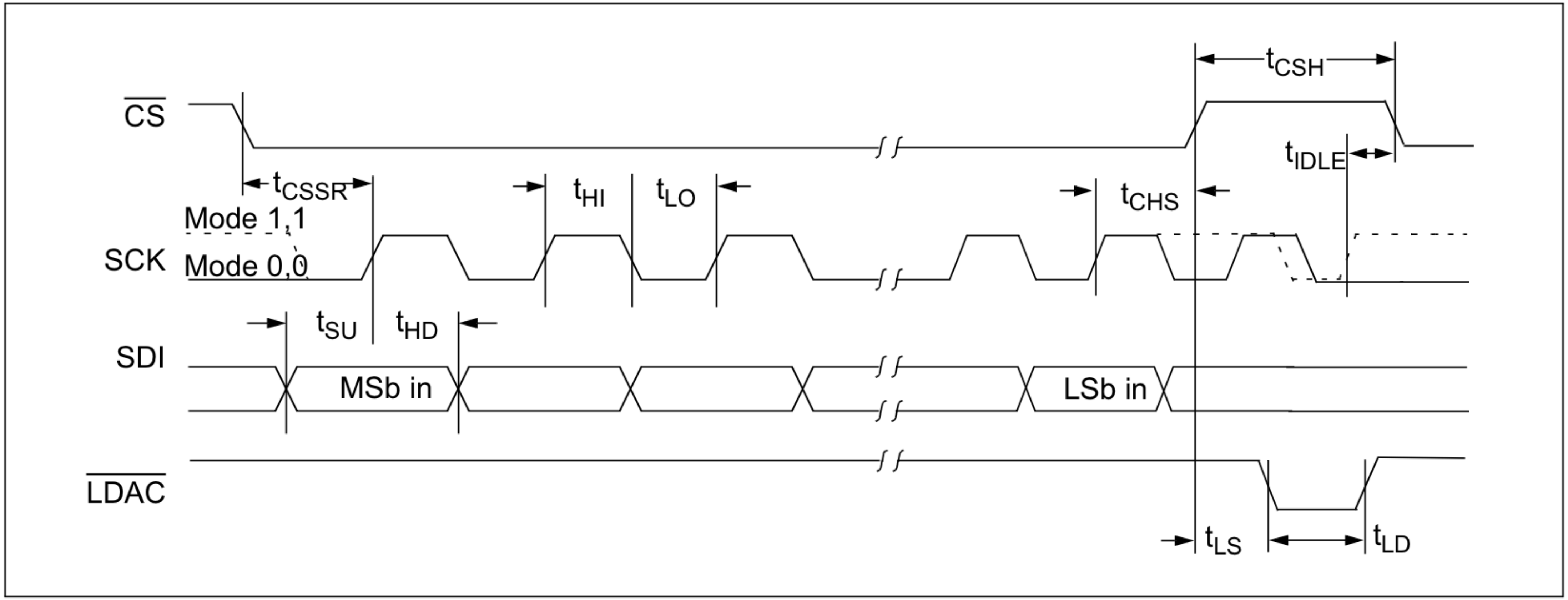


FIGURE 1-1: *SPI Input Timing Data.*

Pin descriptions

3.0 PIN DESCRIPTIONS

The descriptions of the pins are listed in Table 3-1.

TABLE 3-1: PIN FUNCTION TABLE FOR MCP4801/4811/4821

MCP4801/4811/4821		Symbol	Description
MSOP, PDIP, SOIC, DFN	DFN		
1	1	V_{DD}	Supply Voltage Input (2.7V to 5.5V)
2	2	$\overline{CS}$	Chip Select Input
3	3	SCK	Serial Clock Input
4	4	SDI	Serial Data Input
5	5	$\overline{LDAC}$	DAC Output Synchronization Input. This pin is used to transfer the input register (DAC settings) to the output register (V_{OUT})
6	6	$\overline{SHDN}$	Hardware Shutdown Input
7	7	V_{SS}	Ground reference point for all circuitry on the device
8	8	V_{OUT}	DAC Analog Output
—	9	EP	Exposed thermal pad. This pad must be connected to V_{SS} in application

DAC resolution

TABLE 4-1: LSb OF EACH DEVICE

Device	Gain Selection	LSb Size
MCP4801 (n = 8)	1x	2.048V/256 = 8 mV
	2x	4.096V/256 = 16 mV
MCP4811 (n = 10)	1x	2.048V/1024 = 2 mV
	2x	4.096V/1024 = 4 mV
MCP4821 (n = 12)	1x	2.048V/4096 = 0.5 mV
	2x	4.096V/4096 = 1 mV

EQUATION 4-1: ANALOG OUTPUT VOLTAGE (V_{OUT})

$$V_{OUT} = \frac{(2.048V \times D_n)}{2^n} \times G$$

Where:

2.048V	=	Internal voltage reference
D_n	=	DAC input code
G	=	Gain selection
	=	2 for $<\overline{GA}>$ bit = 0
	=	1 for $<\overline{GA}>$ bit = 1
n	=	DAC Resolution
	=	8 for MCP4801
	=	10 for MCP4811
	=	12 for MCP4821

Serial Interface: Write Command

- Initialized by driving NCS pin low, followed by clocking four configuration bits and the 12 data bits on the rising edge of the SCK
- NCS pin then driven high

NCS=$\overline{CS}$

Write Command

REGISTER 5-3: WRITE COMMAND REGISTER FOR MCP4801 (8-BIT DAC)

W-x	W-x	W-x	W-0	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x
0	—	$\overline{GA}$	$\overline{SHDN}$	D7	D6	D5	D4	D3	D2	D1	D0	x	x	x	x
bit 15															bit 0

Where:

bit 15 (1)
0 = Write to DAC register
1 = Ignore this command

bit 14 — Don't Care

bit 13 **$\overline{GA}$:** Output Gain Selection bit
1 = 1x ($V_{OUT} = V_{REF}$ * D/4096)
0 = 2x ($V_{OUT} = 2 * V_{REF}$ * D/4096), where internal V_{REF} = 2.048V.

bit 12 **$\overline{SHDN}$:** Output Shutdown Control bit
1 = Active mode operation. VOUT is available.
0 = Shutdown the device. Analog output is not available. V_{OUT} pin is connected to 500 kΩ (typical).

bit 11-0 **D11:D0:** DAC Input Data bits. Bit x is ignored.

Legend			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	1 = bit is set	0 = bit is cleared	x = bit is unknown

Lab 4 Hints and Suggestions

- Configure SPI peripheral on STM32 MCU and use ADALM2000 to verify data packets
- Build circuit with MCP4801 on breadboard and double check all connections
  - Power with 3.3V!
- With the logic analyzer still connected, connect the oscilloscope to the DAC output and generate output signals

Learning Objectives

By the end of this lab you will have:

- Written C libraries to implement the SPI functionality of the STM32F401RE MCU.
- Interfaced with a digital to analog convertor (DAC) module over an SPI link.
- Written functions to use the DAC to output sine and square waves at a user-specified frequency.
- Used the logic analyzer functionality on the ADALM2000 USB oscilloscope to probe and debug serial data transmission and SPI data packets.

Summary

- Serial interfaces allow us to transfer data between peripherals with only a small number of wires – valuable when we have a limited number of physical pins!
- Serial Peripheral Interface (SPI) is a popular 4-wire, synchronous serial interface which enables full-duplex communication
- SPI can be used to interface with many different peripherals – in Lab 4 you will use it to communicate with a DAC.

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

µP

Feedback