Review Questions: Serial Protocols - SPI, I^2C

SPI, I^2C

1. Is the I^2C bus asynchronous?
   - No. I^2C is a synchronous serial protocol.

2. Do I^2C support multiple Masters?
   - Yes, it supports multiple Masters & multiple Slaves.
   - However SPI do not. SPI supports only one Master.

3. How many I^2C slave can be supported in a 7-bit address scheme?
   - With 7 bits, a total of 128 addresses are possible.
   - However, some addresses are reserved. 2 groups of 8 addresses (0000 XXX & 1111 XXX) are reserved.
   - However, if it is known that the reserved address is never going to be used for its intended purpose, a reserved address can be used for a slave address.
   - *(see next slide)*

SPI, I²C

Table 3. Reserved addresses
X = don't care; 1 = HIGH; 0 = LOW.

Slave address	R/$\overline{W}$ bit	Description
0000 000	0	general call address[1]
0000 000	1	START byte[2]
0000 001	X	CBUS address[3]
0000 010	X	reserved for different bus format[4]
0000 011	X	reserved for future purposes
0000 1XX	X	Hs-mode master code
1111 1XX	1	device ID
1111 0XX	X	10-bit slave addressing

Above table shows the reserved Slave addresses.

Total of 16 addresses are reserved. If used, it reduces the total number of slaves to (128 – 16) = 112.

4. What is the defined clock speed for Standard and Fast modes?
   - Standard mode: clock speed up to 100K bits/s
   - Fast Mode: clock speed up to 400 Kbits/s

SPI, I^2C

5. Does the I^2C clock frequency need to be exactly at the defined Standard or Fast modes?
   - No. However it is important that the devices all operate at the same clock speed.

6. What are the possible I^2C Modes? Name them and describe them briefly.
   - Master-sender: Module issues START and ADDRESS, and then transmits data to the addressed slave device.
   - Master-receiver: Module issues START and ADDRESS, and receives data from the addressed slave device.
   - Slave-sender: Another master issues START and ADDRESS of this module, which then sends data to the master.
   - Slave-receiver: Another master issues START and ADDRESS of this module, which then receives data from the master.

SPI, I^2C

7. Explain how **Repeated START** is used. What situations warrants use of Repeated START?
   - In a normal I^2C transaction, a STOP condition is sent after a Master sends a byte.
   - However it is possible for the Master to send another START condition instead of sending a STOP.
     - In this case, the Master do not release the bus and is able to continue sending data through issuing **Repeated START**.
   - The purpose of **Repeated START** is to allow for combined Write/Read operations to one or more Slaves without releasing the I^2C bus.

SPI, I²C

8. For the Tiva LaunchPad, what value of **TPR** must be programmed to the **I2CMTPR** register to implement an I²C bus speed of 400 KHz (Fast mode)? Assume system bus speed to be 40 MHz.

$$SCL_{Period} = 2 \times (1 + TPR) \times (SCL_{LP} + SCL_{HP}) \times SysClk\_PRD$$

$$SCL_{Period} = \frac{1}{400\ KHz} = 2 \times (1 + TPR) \times (6 + 4) \times 25ns$$

$$TPR + 1 = \frac{1}{(400KHz \times 2 \times 10 \times 25ns)} = 5$$

$$TPR = 5 - 1 = 4$$

We load ***TPR***=*0x04* and set **HS** = '0' in **I2CMTPR** register to obtain an I²C bus speed of 100KHz (Standard mode).

SPI, I²C

9. For the Tiva LaunchPad, what value of **TPR** must be programmed to the **I2CMTPR** register to implement an I²C bus speed of 100 KHz (Standard mode)? Assume system bus speed to be 80 MHz.

$$SCL_{Period} = 2 \times (1 + TPR) \times (SCL_{LP} + SCL_{HP}) \times SysClk\_PRD$$

$$SCL_{Period} = \frac{1}{100\ KHz} = 2 \times (1 + TPR) \times (6 + 4) \times 12.5ns$$

$$TPR + 1 = \frac{1}{(100KHz \times 2 \times 10 \times 12.5ns)} = 40$$

$$TPR = 40 - 1 = 39$$

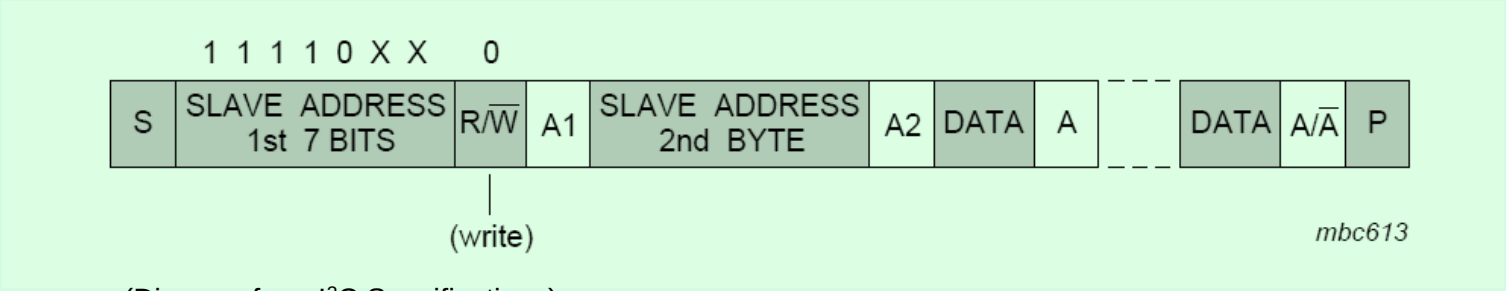
We load ***TPR***=*0x27* and set **HS** = '0' in **I2CMTPR** register to obtain an I²C bus speed of 100 KHz (Standard mode).

SPI, I^2C

10. When 2 I^2C Masters are trying to send data to the bus, how do they each Master determine if it has control of the bus and thus able to send? *[Arbitration process]*
    - Arbitration proceeds **bit by bit**. During every bit transfer, each Master checks to see if the SDA level matches what it has sent.
    - When a Master tries to send a 'H', but detects that the SDA level is 'L', the master knows that it has lost the arbitration & turns off its SDA output driver. The other Master goes on to complete its transaction.

SPI, I²C

11. I²C protocol can support **10-bit addresses**. Find out how 10 bit addresses are sent on the SDA signal.
    - 10-bit slave address is formed from the first two bytes following a **START** condition or **Repeated START** condition.
    - The 1st 7 bits of the first byte are the combination 1111 0xx of which the last 2 bits (xx) are the 2 Most-Significant Bits (MSB) of the 10-bit address; the 8th bit of the 1st byte is the R/W bit that determines the direction of the message.

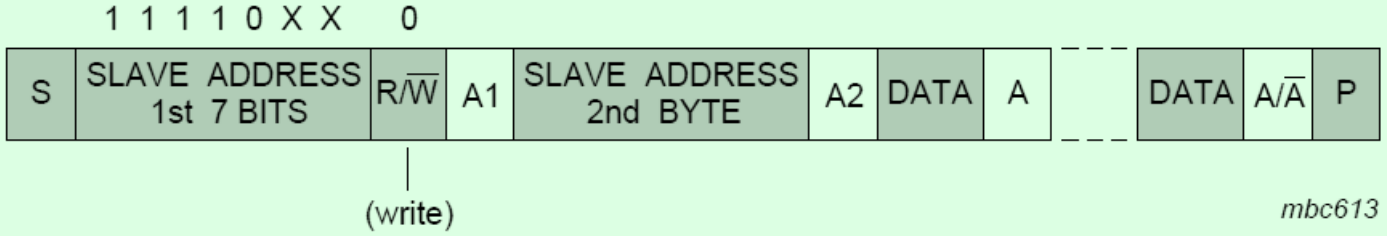


(Diagram from I²C Specifications)

SPI, I²C

Master-Transmitter:

- When a 10-bit address follows a **START** condition, each slave compares the 1st 7 bits of the 1st byte of the slave address (1111 0xx) with its own address & tests if the eighth bit (R/W) is 0.
  - It is possible that more than one device finds a match and generate an **ACK** (A1).
- Slaves that found a match compare the 8 bits of the 2nd byte of the slave address with their own addresses.
- The Slave that finds an address match will generate an **ACK** (A2).
- The matching slave remains addressed by the master until it receives a **STOP** (P) or **Repeated START** condition followed by a different slave address.

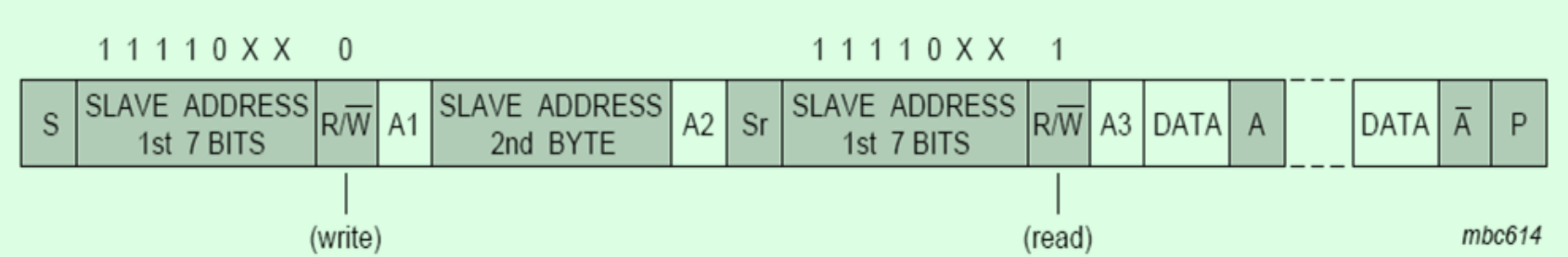


(Diagram from I²C Specifications)

SPI, I²C

Master-Receiver:

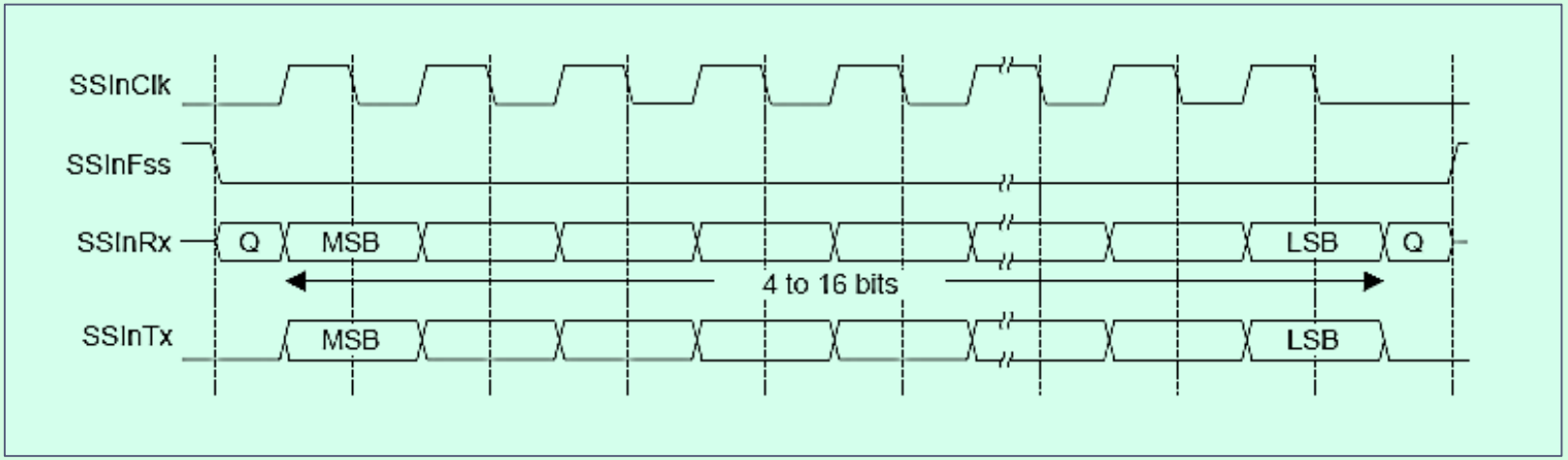
- Transfer direction is changed after the 2nd R/W bit.
- Up to **ACK** bit A2, bus transaction is same as for a Master-Transmitter addressing a slave-receiver.
- After the Repeated START (Sr), a matching slave remembers that it was addressed before. This slave then checks if the 1st 7 bits of the 1st byte of slave address following Sr are the same, & tests if the 8th (R/W) bit is 1.
- If there is a match, the slave considers that it has been addressed as a transmitter & generates **ACK** A3. The slave-transmitter remains addressed until it receives a **STOP** (P) or another **Repeated START** condition (Sr) followed by a different slave address.



(Diagram from I²C Specifications)

SPI, I^2C

12. Which **SPI Mode** is the following SPI transaction? Reference the Mode in terms of the SPO and SPH bit logic levels.
    - Clock is low when not clocking => CPOL = 0;
    - Data is clock at 2nd clock transition => CPHA = 1
    - Therefore, **SPI Mode = 1**.



SPI, I^2C

13. What are the 4 SPI modes? List them and describe their differences.
    - **Mode 0:** Clock Polarity is 'L', Data is clocked in during 1^{st} clock transition.
    - **Mode 1:** Clock Polarity is 'L', Data is clocked in during 2^{nd} clock transition.
    - **Mode 2:** Clock Polarity is 'H', Data is clocked in during 1^{st} clock transition.
    - **Mode 3:** Clock Polarity is 'H', Data is clock in during 2^{nd} clock transition.

SPI, I^2C

14. Does each SPI Slave has individual address?
    - No. SPI does not have the concept of addresses, unlike I2C. Each slave is selected through the SS pin.

15. How many SSI modules do the Tiva LaunchPad has? Name them.
    - Total of 4 modules: SSI0, SSI1, SSI2, SSI3.

16. Which register and bit is used to enable a SPI module?
    - SSE bit of the SSI Control Register 1 (SSICR1).

17. Which register is used to enable the clock to a SSI module? What would be a reason for implementing such a feature?
    - The clock to the SSI module is enabled through the SSI Clock Gating Register (RCGCSSI).
    - The ability to disable a clock the SSI module (or other peripheral modules) is for power management. Many embedded systems are battery powered and therefore require power management features to maximize battery life.

SPI, I^2C

18. Which SSI register is used to check if a SSI module's register is ready for Read/Write operations?
    - A SSI module is ready if the module's bit in the SSI Peripheral Ready Register (PRSSI) is set.

19. If you wish to set the SSI module for SPI clock of 4 MHz transmit/receive rates, what values of **CPSDVSR** & **SCR** would you use?

    Bit Rate (BR) = SysClk/(CPSDVSR * (1 + SCR))
    = 80,000,000 /(2 * (1 + 9) - assume 80MHz bus clock
    CPSDVSR = 0x02 *(must be even)*
    SCR = 9

20. What is the base address for the SSI3 module?
    - 0x4000.B000.