SPI (Serial Peripheral Protocol)

- Developed by Motorola for short distance commn.

Bus protocol

- SPI is bus protocol i.e. multiple devices can be connected on a single bus.
- SPI is multi-master bus protocol i.e. one bus can have multiple devices which can become master. However only one master at a time.
- Master generates clock and select (enable) the slave device to be communicated.

Physical characterics

- 4 wire bus protocol
- 4 wires
    - SCK/SCLK
    - MOSI/SDO
    - MISO/SDI
    - SS/CE
- Full duplex protocol
- Internally there is a single shift register used for tx & rx.
- TTL voltage levels

Logical characterics

- CPOL (Clock Polarity)
    - CPOL=0: Clock base is 0 and first edge is rising.
    - CPOL=1: Clock base is 1 and first edge is falling.
- CPHA (Clock Phase)
    - CPHA=0: Data is sampled/read on leading edge.
    - CPHA=1: Data is sampled/read on trailing edge.

CPOL	CPHA	Data reading
0	0	Rising
0	1	Falling
1	0	Falling
1	1	Rising

- Data (single byte) write/read
    - Select the slave (SS=0)
    - Write/Read a byte of data (DR)
    - Deselect the slave (SS=1)

- Data (multi byte) write
    - Select the slave (SS=0)
    - Write internal address of slave
    - Write bytes of data (DR)
    - Deselect the slave (SS=1)
- SPI errors
    - Write Collision
        - Writing new data before current data is transmitted.
    - Read Overrun
        - Data is not read and new data arrive.
    - Mode Fault
        - Switch from Master mode to Slave mode when SS pin of current master device is cleared by some other device on bus.
    - Slave Abort
        - When slave device is not responding to master.

Applications

- SPI enabled LCD, RTC, EEPROM, ...
- SD-Card
- SPI-CAN bus (MCP-2515)

LPC1768 SPI

- Legacy SPI
    - SPI0
    - SPI is protocol.
- SSP (Serial Synchronous Port)
    - SSP0, SSP1.
    - SSP is 4-wire port that supports multiple protocols.
        - SPI - Motorola
        - SSI - TI
        - Microwire - NS
    - 8-frame FIFO

BlueBoard SPI cct

- SSP0 --> Shift register (74HC595)
    - (P0.15) SCK --> SH_CP (Shift Clock) - Rising edge

    - (P0.16) SS (GPIO) --> ST_CP (Storage Clock Pulse) - Rising edge
    - (P0.18) MOSI --> DS (Serial Data - IN)
    - (P0.17) MISO --> Q7' (Serial Data Out)

SPI/SSP Programming

- Pin selection
    - P0.15 as SCK0
        - PINSEL0[31:30] = 10
    - P0.17/P0.18 as MISO & MOSI
        - PINSEL1[3:2] & PINSEL1[5:4] = 10
    - P0.16 as GPIO (output)
        - PINSEL1[1:0] = 00
        - FIODIR to set as output
- CPSR register
    - Will divide PCLK to generate SPI clock.
    - Should be even number between 2 to 254.
    - Config: CPSR=2
        - PCLK = 18 MHz
        - SPI clock = 9 MHz
- CR0 register
    - bits[3:0] = 0111 for 8-bit transfer
    - bits[5:4] = 00 for SPI frame format
    - bit [6] = CPOL=1
    - bit [7] = CPHA=1
    - bits[15:8] = SCR
        - SPI clock = PCLK / (CPSDVSR * [SCR+1])
            - CPSDVSR = 2 (set in CPSR)
            - SCR = 0x00 (default)
            - SPI clock = 9 MHz
- CR1 register
    - bit[1] = SSP Enable(1) or Disable(0)
    - bit[2] = Master(0) or Slave(1)
- DR register
    - 4 to 16 bits
- SR register
    - bit[0] = Tx FIFO empty(1) or not empty(0).
    - bit[2] = Rx FIFO not empty(1) or empty(0).