Design of an SPI receiver

(Péter Szántó, Péter Csordás – BME MIT)

In the first part, a generic SPI receiver design is described. In the second part, the most important parameters of the TMP121 temperature sensor SPI interface is given. Before reading this paper, review the paper about Serial Communication Standards.

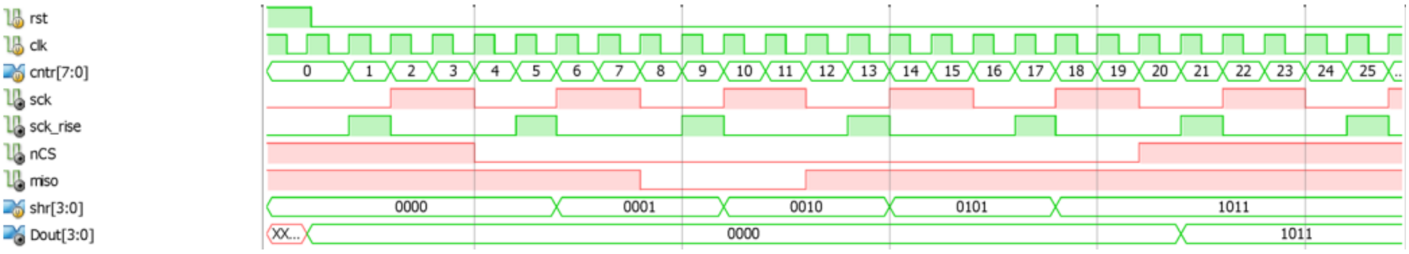
Generic SPI receiver

The interface and the tasks of an SPI receiver:

- Ensure the SPI clock with the appropriate frequency (**SCK** output)
- Select the SPI slave device by setting the Chip Selected inverted signal (**nCS** output) to 0.
- Sample the Master In Slave Out data line (**MISO** input) at SCK edges.
- Store the read data for further processing (**Dout** bus output)
- The module uses the system clock and reset signal (**clk** and **rst** input).

During a read cycle, the SPI clock must be ensured continuously. At the beginning, the slave must be selected. The specified number of bits must be read, and the slave has to be deselected. Before the next reading, the slave must be kept deselected for a predefined time (given in the slave specification.). The Dout output must be consistent, thus during the data read process, the previous value must be held.

The design is given parametric. The following figure illustrates the functionality for p=1, r=r2=4. (See the text below.)

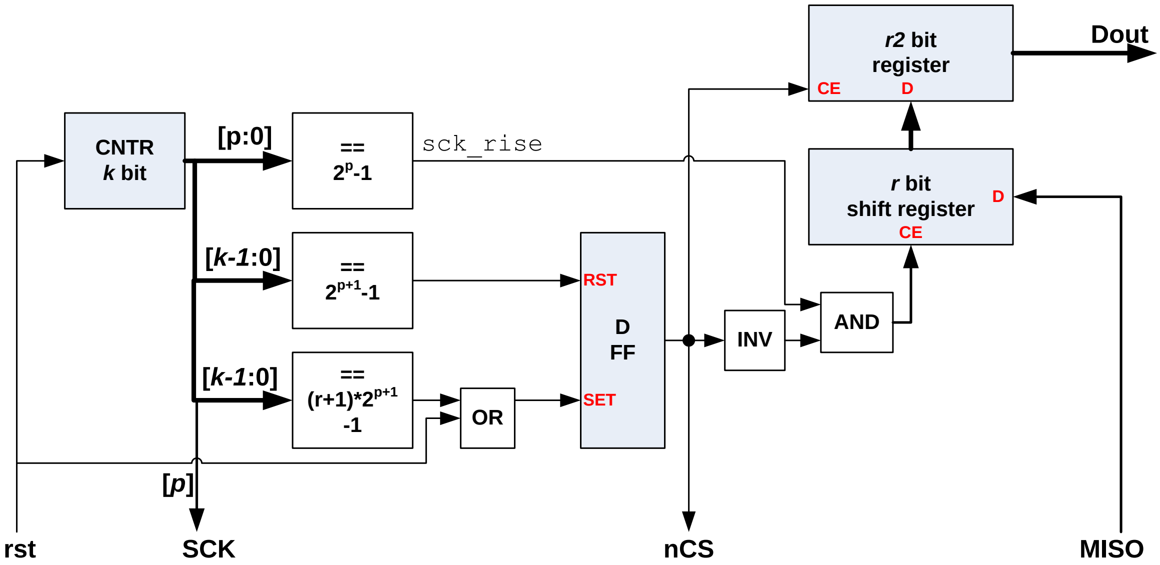


Considerations for the design:

- The total receiving cycle (including the wait period) can be scheduled by means of an **internal counter (cntr).** The length of the cycle can be given the period of the counter. (We do not have to read the data with exactly the maximal possible rate, some extra wait time is allowed.)
- The SPI communication should be done at the beginning of the counter cycle. If the communication time is much shorter than the t$_w$ wait period, **the bit-wide (*k*) of the counter** is determined by t$_w$ and the system clock frequency (f$_{clk}$): $k = \lceil \log_2(f_{clk} * t_w) \rceil$
- The slave specification allows a wide range for SCK. Thus, it is always possible, to derive SCK as system clock divided by a power of 2: $f_{SCK} = f_{clk}/2^{(p+1)}$. Thus, **SCK is given as the pth bit of cntr**. (In the depicted example p=1)

- We can assume, that the slave writes the MISO line at falling SCK edges, thus it should be sampled at rising edges. The **sck_rise is an enable signal for sampling**. The sampling is enabled, if the last p+1 bits of the counter has a value of $2^p - 1$. (In the depicted example, the last counter bits are 2'b01) In this case, with the *next* system clock rising edge we will have an SCK rising edge and the sampling will be enabled.
- The slave specifies that it must be selected at least with a predefined time before reading the first bit. This time is usually shorter than the half period of SCK, thus **falling of nCS can be synchronized to the first SCK falling edge of the cycle**. This edge comes after $cntr = 2^{p+1} - 1$ (=3 in our example).
- After reading the r data bits sent by the slave to **an r bit shift register**, the slave should be deselected. The **deselection of the slave** can be synchronized to the (r+1)th falling edge of SCK, thus to cntr = $(2^{p+1}) * (r + 1) - 1$. (In the depicted example nCS rises after cntr=19.)
- To keep the module output consistent, the relevant valid data bits should be copied to a register while no communication is in progress, thus if nCS=1.

This considerations lead to the following block diagram:

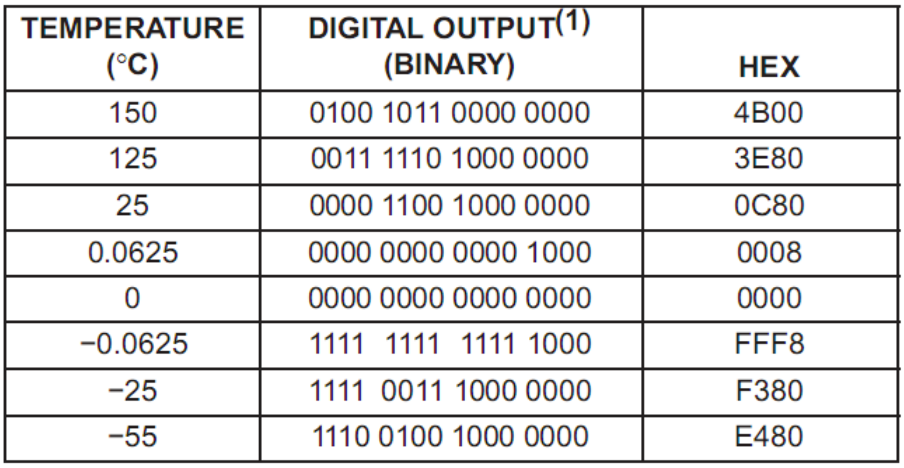


Notations:

- Grey boxes: Sequential logic. All FF are running from the main 50 MHz clock. (Not shown.)
- White boxes: Combinatorial logic
- Red texts: Inputs of the functional block
  - RST: FF reset input → clears the FF into logic low.
  - SET: set input → sets the FF into logic low.
  - CE: FF clock enable input.
  - D: FF data input.
- SCK: SPI clock output. (Output of the FPGA)
- nCS: SPI Chip Select. (Output of the FPGA)
- MISO: SPI Serial Data Output. (Input of the FPGA)
- *p, k, r* and *r2* are parameters.

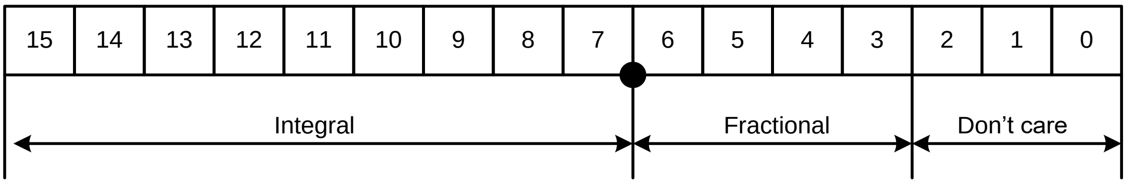
The TMP121 data format and SPI interface

The serial data of the TMP121 consists of 12-bit plus sign temperature data (sum 13 bits) followed by three high-impedance (don't care) bits. (All together 16 bits) Data is transmitted in Binary Two's Complement format. Next figure describes the output data of the TMP121.

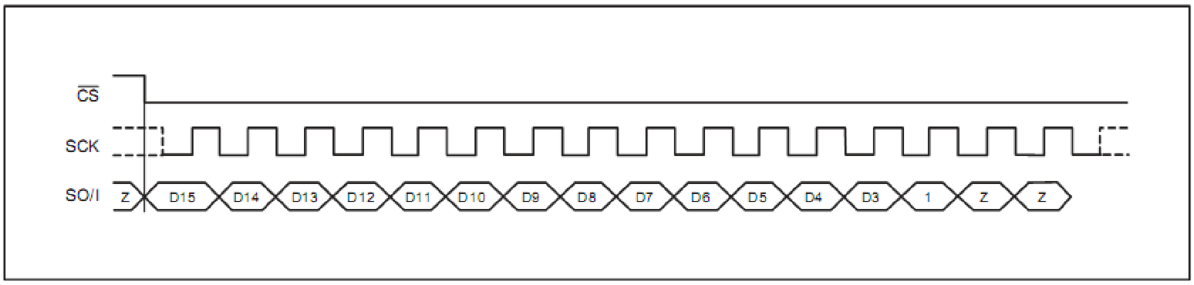


TEMPERATURE (°C)	DIGITAL OUTPUT$^{(1)}$ (BINARY)	HEX
150	0100 1011 0000 0000	4B00
125	0011 1110 1000 0000	3E80
25	0000 1100 1000 0000	0C80
0.0625	0000 0000 0000 1000	0008
0	0000 0000 0000 0000	0000
−0.0625	1111 1111 1111 1000	FFF8
−25	1111 0011 1000 0000	F380
−55	1110 0100 1000 0000	E480

It can be seen that the change in least significant bit, responds to 0.0625 °C temperature change. As 1/0.0625 = 16, the fractional part can be stored on 4 bits, while the integral part is (13-4=9) 9 bits in width.

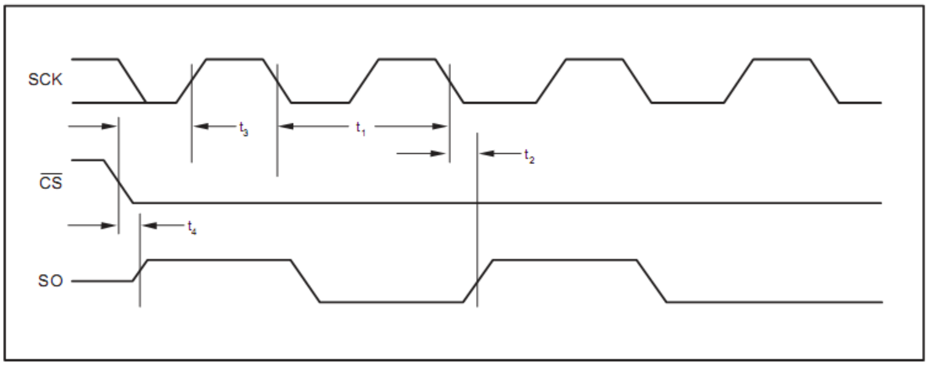


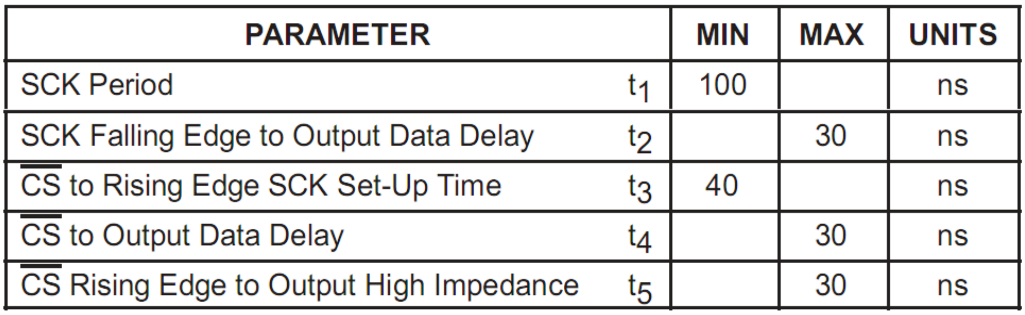
The TMP121 uses SPI-compatible interface. Following figures describe the various timing requirements, with parameters defined in Table.



Transfer start is indicated by pulling the **Chip Select (CS)** low. Data is shifted out into wire **Serial Output (SO)** in the following 16 clock cycles (**SCK**). The 16 bit format is the one described above; the last 3 bits are don't care. The bits are shifted out on the **SCK falling-edge**.

The timing requirements (from the datasheet) as follows:





PARAMETER		MIN	MAX	UNITS
SCK Period	t_1	100		ns
SCK Falling Edge to Output Data Delay	t_2		30	ns
$\overline{\text{CS}}$ to Rising Edge SCK Set-Up Time	t_3	40		ns
$\overline{\text{CS}}$ to Output Data Delay	t_4		30	ns
$\overline{\text{CS}}$ Rising Edge to Output High Impedance	t_5		30	ns

As a summary:

- **t1**: The period of SCK is minimum 100 ns. (Therefore the max. frequency is 10 MHz).
- **t2**: SCK Falling Edge to Output Data Delay is 30 ns. As a conclusion data should not be sampled earlier by the FPGA.
- **t3**: There must be a 40ns delay between the falling edge of CS, and the rising edge of SCK.
- **t4**: The data appears on the output after CS falling edge not later than 30 ns.
- **t5**: After the riding edge of CS, data line is driven active at maximum 30 ns.

The conversion of the temperature occurs while the sensor is unselected. The minimal time for a conversion is **320 ms**.