Interfacing ADC with Krypton

Aim

We wish to interface ADC0804 and MCP3008 with Krypton.

Theory

Analog-to-Digital converter (ADC) is a device which converts analog signal into an equivalent digital signal. The basic blocks of an ADC are sample and hold block, quantization block and encoder block. The block diagram of an ADC is shown in the figure 1. Let us now look at the working of two available ADCs.

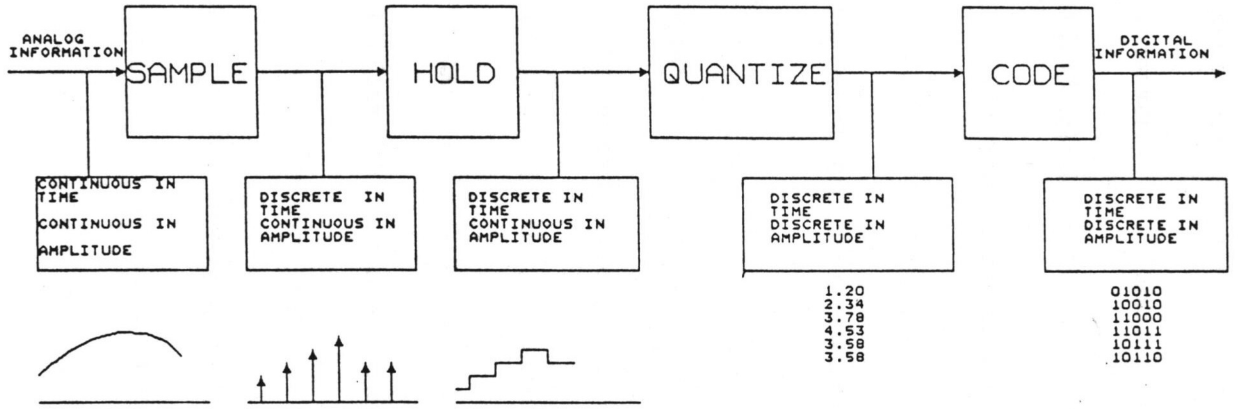

ANALOG INFORMATION
SAMPLE
HOLD
QUANTIZE
CODE
DIGITAL INFORMATION
CONTINUOUS IN TIME CONTINUOUS IN AMPLITUDE
DISCRETE IN TIME CONTINUOUS IN AMPLITUDE
DISCRETE IN TIME CONTINUOUS IN AMPLITUDE
DISCRETE IN TIME DISCRETE IN AMPLITUDE
DISCRETE IN TIME DISCRETE IN AMPLITUDE
1.20
2.34
3.78
4.53
3.58
3.58
01010
10010
11000
11011
10111
10110

Figure 1: ADC Block Diagram(Courtesy: CorrelatorSystemSchematics.com)

The first ADC IC that we are interested in is ADC0804 which is a parallel ADC i.e. the encoded bits are available at the output as 8 parallel bits. The second one is ADC MCP3008 which is a serial ADC where the output is made available as a stream of bits.

To begin with let us discuss ADC0804. ADC0804 is a CMOS 8-bit successive approximation analog to digital converter. It operates at a supply voltage of 5 V. The minimum and maximum operating clock frequencies are 100 kHz and 1460 kHz respectively. Typically it is advised to operate the IC at 640 kHz which is specified in the data sheet by the manufacturer. At higher clock frequencies accuracy can degrade. For lower clock frequencies, the duty cycle limits can be extended so long as the minimum clock high time interval or minimum clock low time interval is no less than 275 ns.

The MCP3008 IC is a successive approximation 10-bit ADC with an on-board sample and hold circuitry. The MCP3008 is programmable to provide four pseudo-differential input pairs or eight single-ended inputs. The device operates over a broad voltage range (2.7V - 5.5V). They are capable of conversion rates of up to 200 kbps. It uses SPI protocol to communicate with the master (or the controlling device). The Serial Peripheral Interface (SPI) protocol operates in full a duplex mode. Devices communicate in master/slave mode where the master device initiates the data frame. Multiple slave devices are allowed with individual slave select lines. Sometimes SPI is called a four-wire serial bus.

Pin and timing diagrams of ADCs

To be able to interface an ADC with the master device (which could be a microcontroller, FPGA or a CPLD) we need to know the pin configuration of the ADC which is being used, the timing diagrams of the ADC and the protocol used by the ADC to communicate with the master. So let us begin by illustrating the pin and timing diagrams of ADC0804. The pin diagram of ADC0804 is shown in figure 2. For understanding the behavior ADC we need to have a thorough information about the timing diagrams of the ADC. The timing diagram diagram of ADC0804 is shown in figure 3. On similar lines the pin diagram of MCP3008 is shown in figure 4. The configuration details in given in figure 5. The pin description is given in figure 6. The timing diagram diagram of MCP3008 is

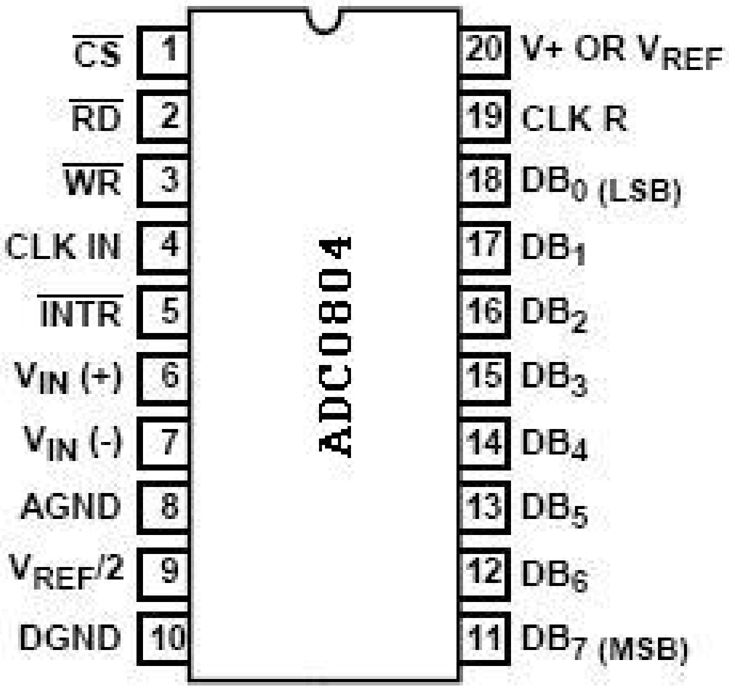


Figure 2: Pin digram of ADC0804

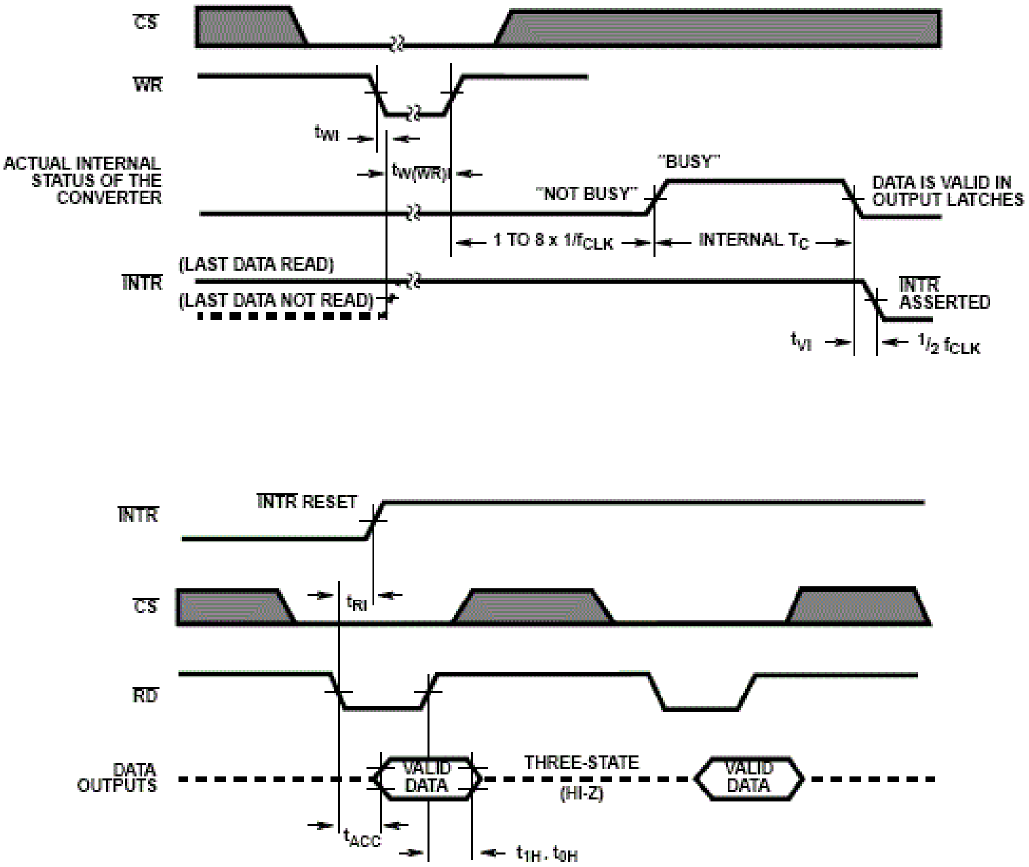


Figure 3: Timing digram of ADC0804

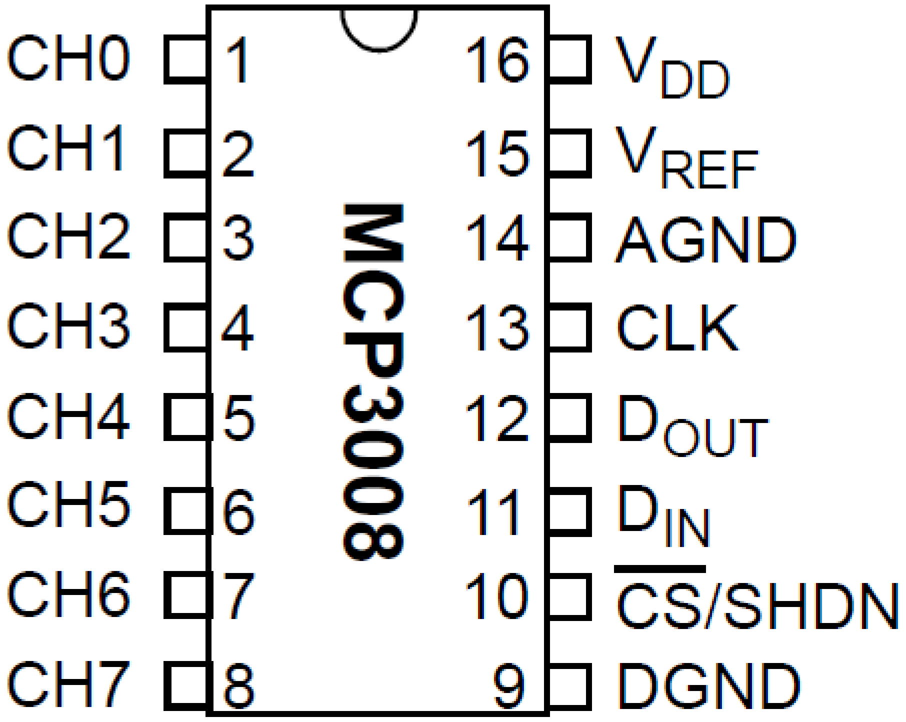
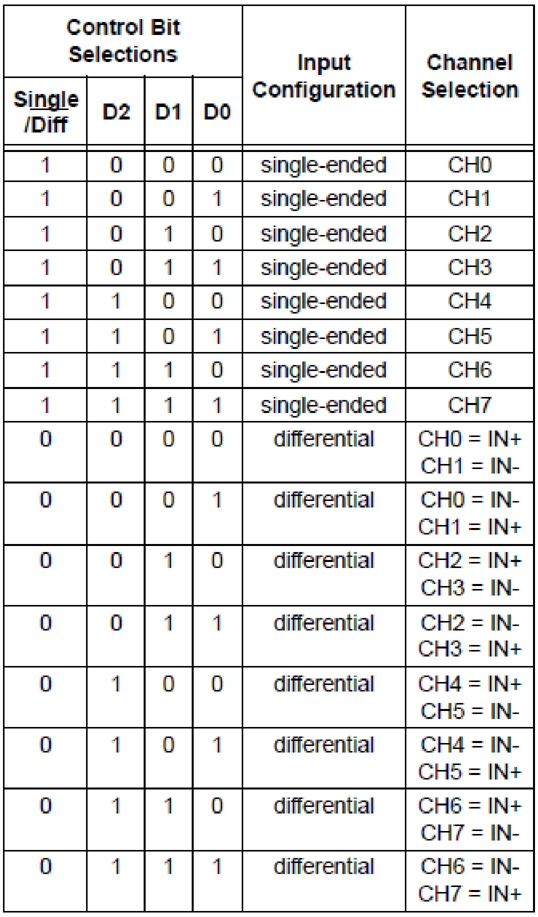


Figure 4: Pin digram of MCP3008

shown in figure 7.



Control Bit Selections				Input Configuration	Channel Selection
Single /Diff	D2	D1	D0		
1	0	0	0	single-ended	CH0
1	0	0	1	single-ended	CH1
1	0	1	0	single-ended	CH2
1	0	1	1	single-ended	CH3
1	1	0	0	single-ended	CH4
1	1	0	1	single-ended	CH5
1	1	1	0	single-ended	CH6
1	1	1	1	single-ended	CH7
0	0	0	0	differential	CH0 = IN+ CH1 = IN-
0	0	0	1	differential	CH0 = IN- CH1 = IN+
0	0	1	0	differential	CH2 = IN+ CH3 = IN-
0	0	1	1	differential	CH2 = IN- CH3 = IN+
0	1	0	0	differential	CH4 = IN+ CH5 = IN-
0	1	0	1	differential	CH4 = IN- CH5 = IN+
0	1	1	0	differential	CH6 = IN+ CH7 = IN-
0	1	1	1	differential	CH6 = IN- CH7 = IN+

Figure 5: Configuration bits table for MCP3008

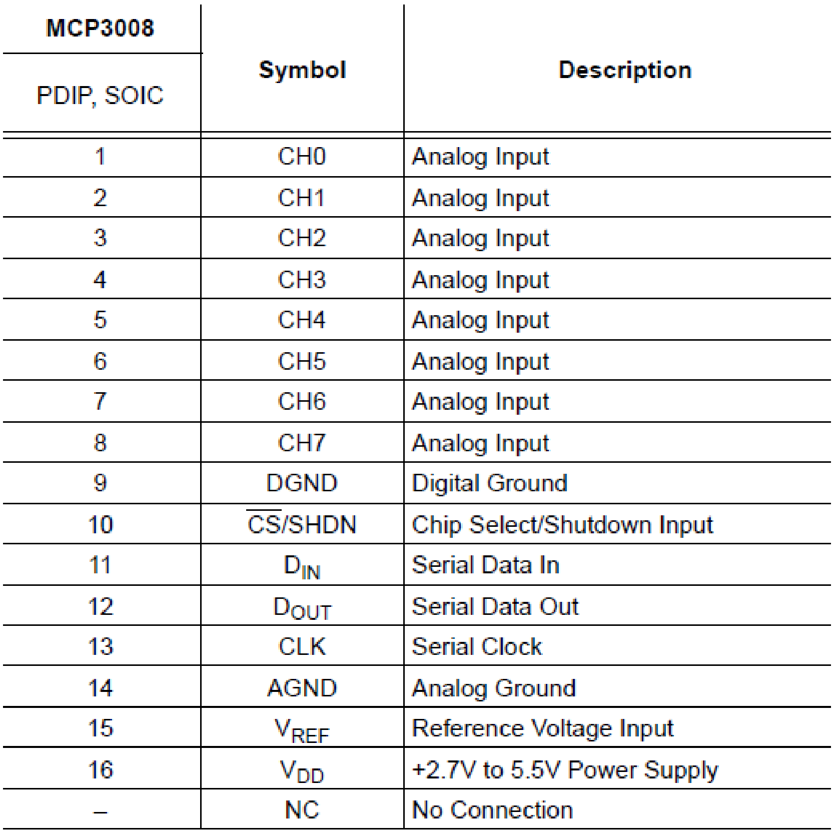
Consolidated problem statement

1. Interface ADC0804, a standard 8-bit ADC with Krypton. Try to generate all control signals for the ADC from Krypton itself. Use the 8 on-board LEDs to indicate the digitized input voltage. Test your circuit by applying a sawtooth waveform from a function generator of frequency 0.5 Hz (so that the changing values are clearly noticeable).

2. For easier use, we wish to interface the ADC MCP3008 with Krypton. This is an SPI interface, so you have to describe an SPI controller in HDL. Test the ADC in a similar way as in (1).

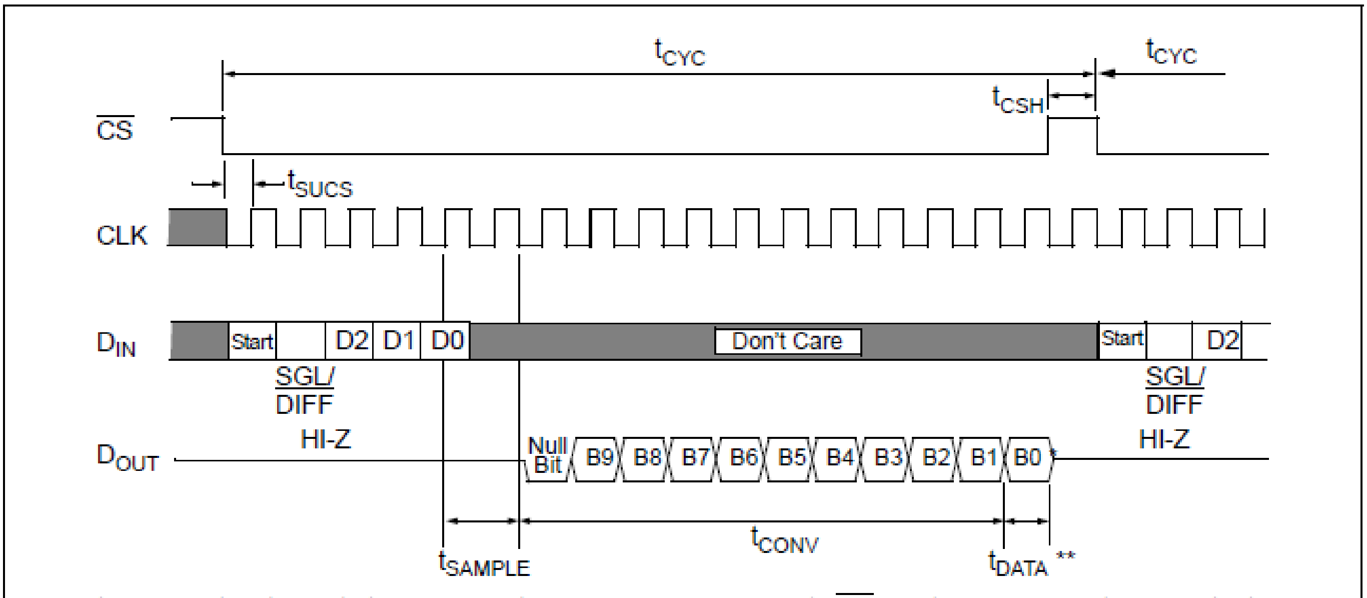
Design flow for ADC0804 controller

Before getting into the design let us recall that while mentioning the operating frequency of ADC it was advised to operate the ADC at optimally 640 kHz. It was also given that we could operate the ADC at lower frequencies as long as a



MCP3008	Symbol	Description
PDIP, SOIC		
1	CH0	Analog Input
2	CH1	Analog Input
3	CH2	Analog Input
4	CH3	Analog Input
5	CH4	Analog Input
6	CH5	Analog Input
7	CH6	Analog Input
8	CH7	Analog Input
9	DGND	Digital Ground
10	$\overline{\text{CS}}$/SHDN	Chip Select/Shutdown Input
11	D_{IN}	Serial Data In
12	D_{OUT}	Serial Data Out
13	CLK	Serial Clock
14	AGND	Analog Ground
15	V_{REF}	Reference Voltage Input
16	V_{DD}	+2.7V to 5.5V Power Supply
–	NC	No Connection

Figure 6: Pin description for MCP3008



* After completing the data transfer, if further clocks are applied with $\overline{\text{CS}}$ low, the A/D converter will output LSB first data, then followed with zeros indefinitely. See Figure 5-2 below.

** t_{DATA}: during this time, the bias current and the comparator powers down while the reference input becomes a high-impedance node.

Figure 7: Timing digram of MCP3008

few conditions mentioned are satisfied. But the Krypton board has a 50 MHz system clock which needs to be divided as shown in equation 1.

$$ADC_{clk} = \frac{50Mhz}{2^8} = 390 \text{ kHz} \tag{1}$$

For interfacing ADC0804 with Krypton board the circuit shown in figure 8 proves to be useful.

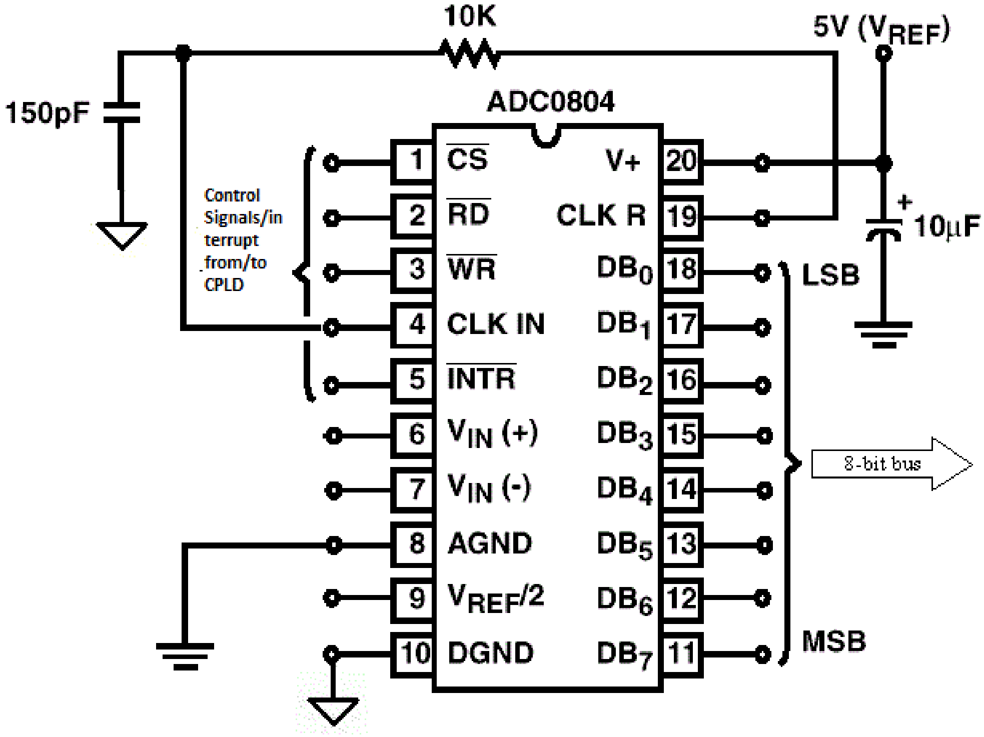


Figure 8: Circuit diagram of ADC0804

State Diagram

The state diagram explaining the logic behind the behavioral description of the master is shown in Fig. 9

Algorithm

The control signals are periodically generated to operate ADC in free running mode. Control outputs from CPLD are `CSbar`, `WRbar` and `RDbar` which represent the chip select, write and the read signals. `INTR` is the interrupt from ADC indicating the completion of conversion. Conversion starts 8 clock cycles after the positive edge of `WRbar` and conversion is completed after `INTR`=0. Valid data is available at the output after there is negative edge on `RDbar`. For clear understanding refer to the timing diagram given in Fig. 3.

- **Idle state**: Here all signals are high and ADC is waiting for `CSbar` to go low.

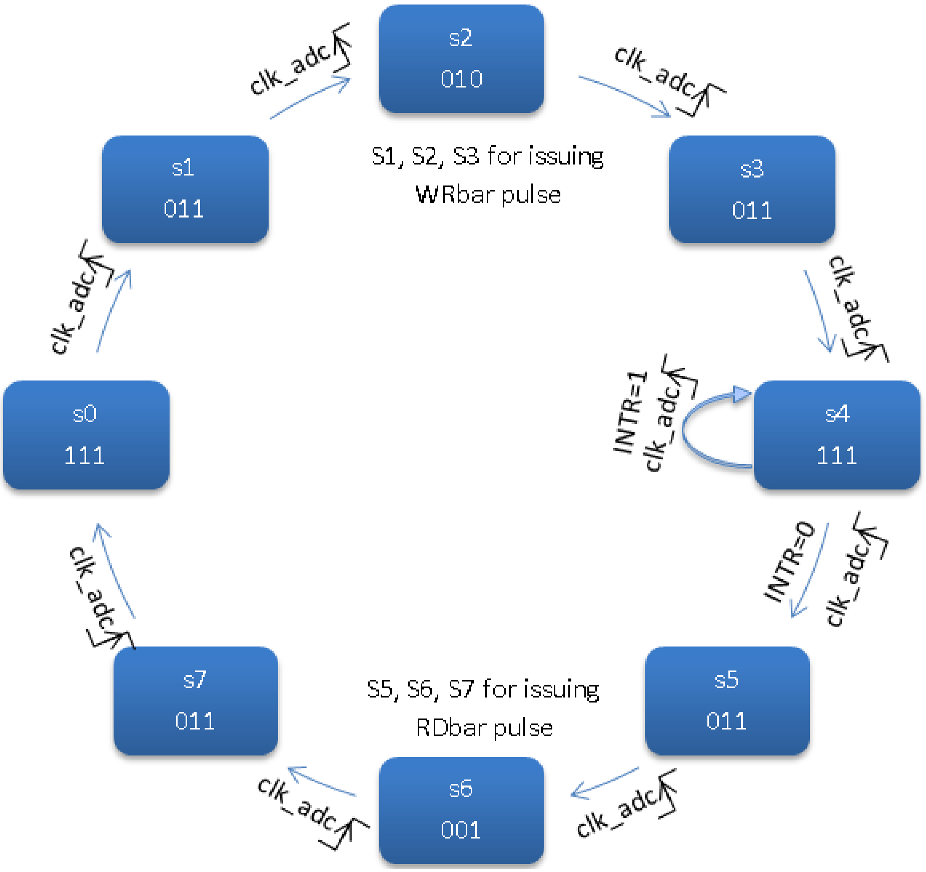


Figure 9: Finite State Machine

- **Chip select**: Chip must be selected before issuing `WRbar`==0;
- **Start conversion**: Positive edge on `WRbar` starts conversion.
- **Wait for conversion**: Wait for conversion to complete i.e. check if `INTR`==0;
- **Reading data**: As soon as data is read by `RDbar`==0; along with `CSbar`==0 for issuing `RDbar` signal, after which `INTR` goes high. Valid data is available only after some time(tacc) after negative edge on `RDbar`. This valid vanishes after finite time after positive edge on RDbar
- **Idle state**: Making `CSbar`==1 makes it idle once again.

Design flow for MCP3008 controller