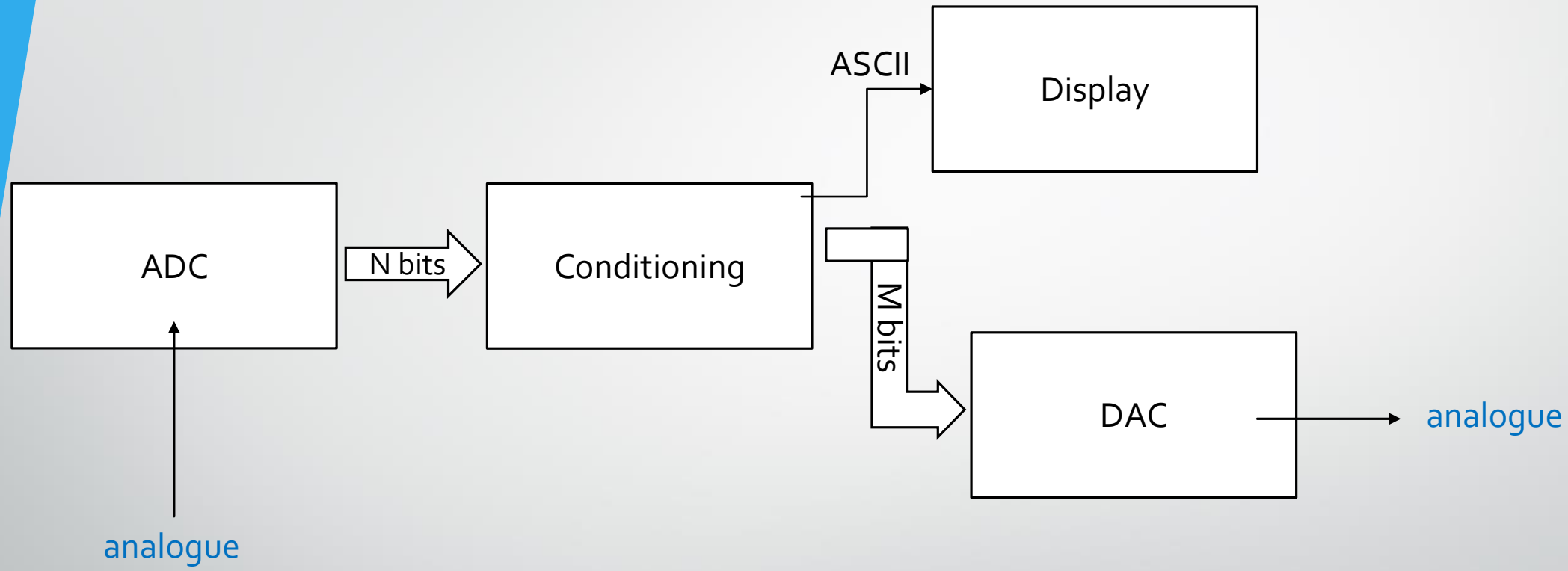
# Attempt to Build a Simple Acquisition System with FPGA

Dawid Pietruch supervisor: dr inż. Andrzej Skoczeń

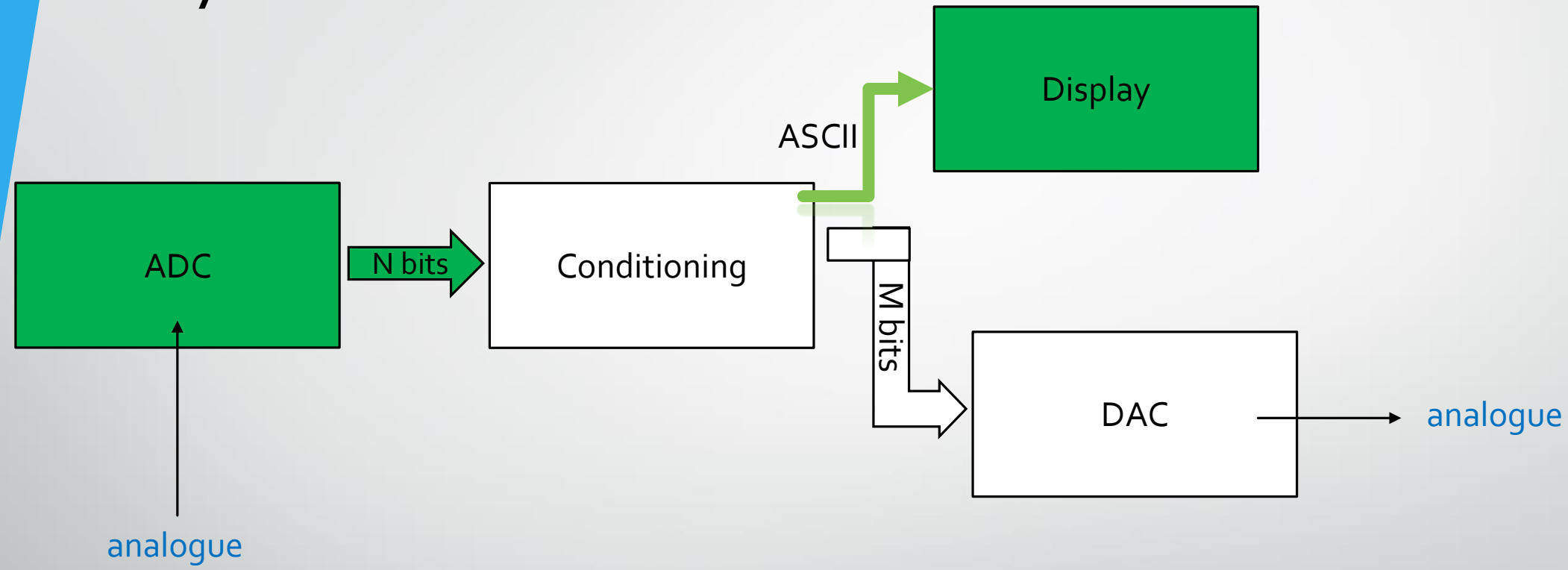
204b72616b6f77204170706c6965642020506879736963732020202020616e6420436f6d70757465722053636369656e636553756d6d6572205363686f6f6c273230

Krakow Applied Physics and Computer Science Summer School '20

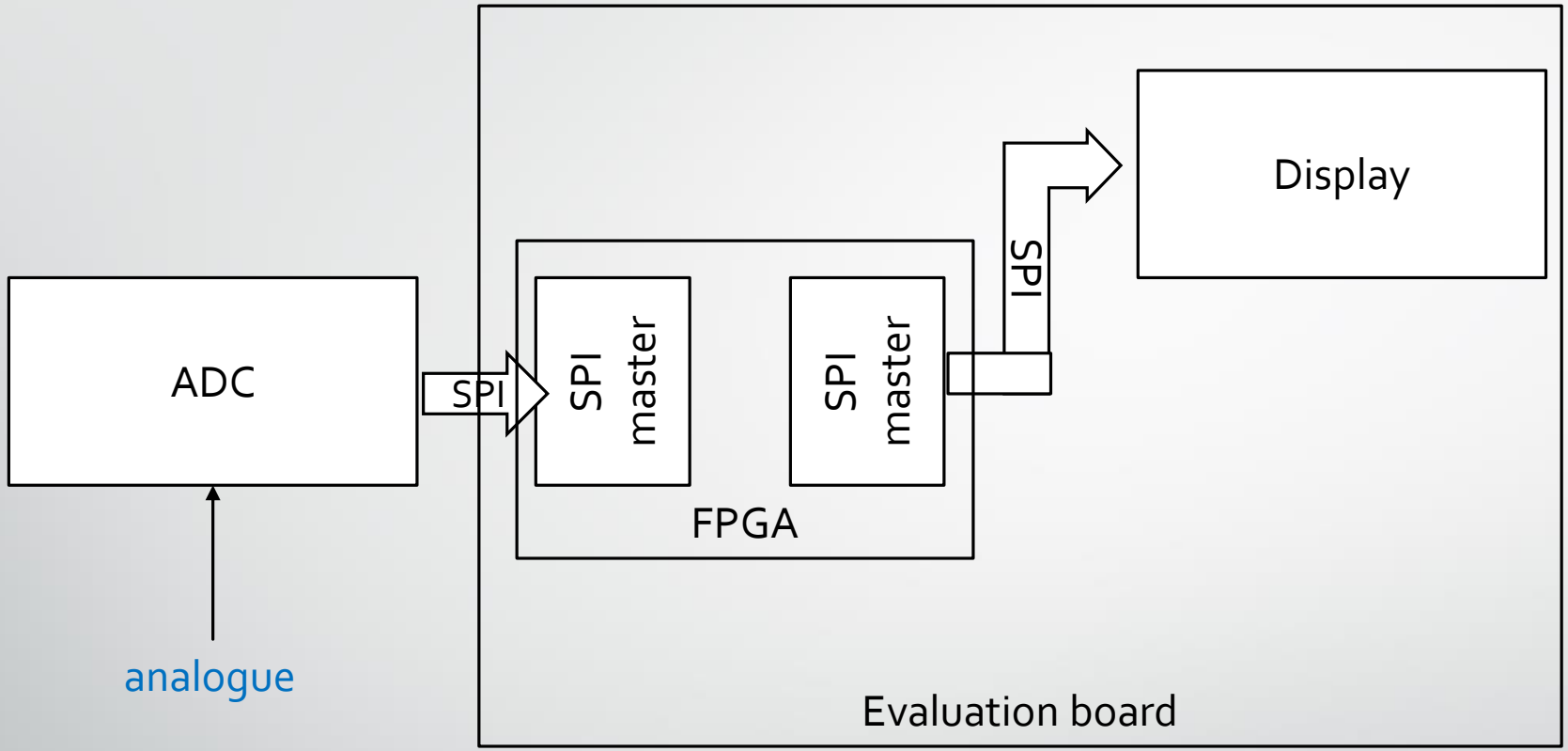
# General idea



# In practice this presentation concerns only some of the modules

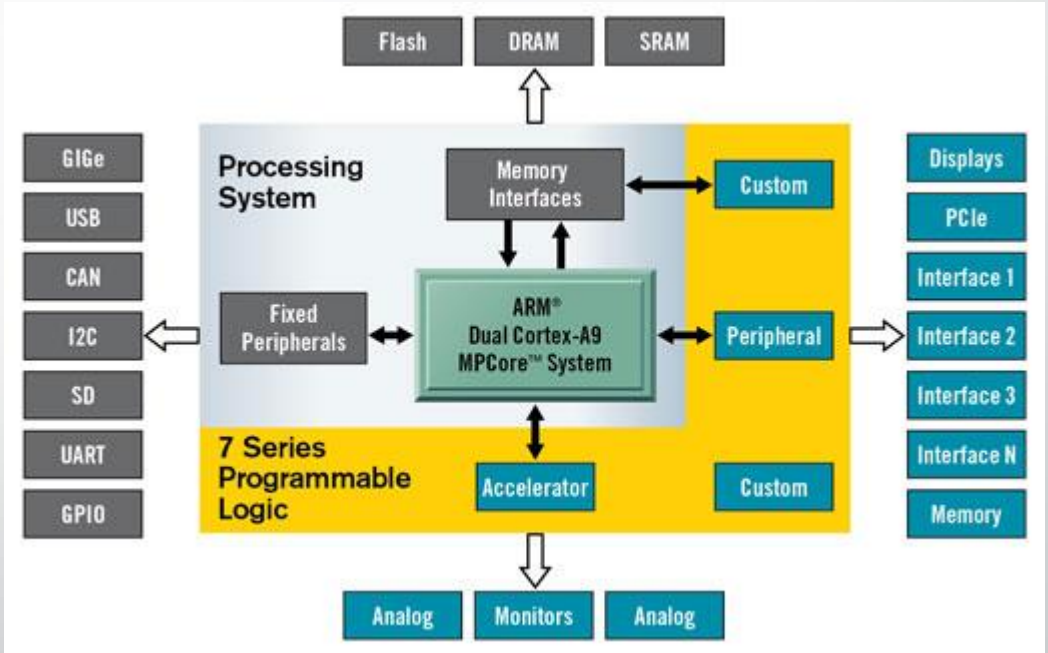


# Current achievement include



# FPGA- *field „programmable" gate array*

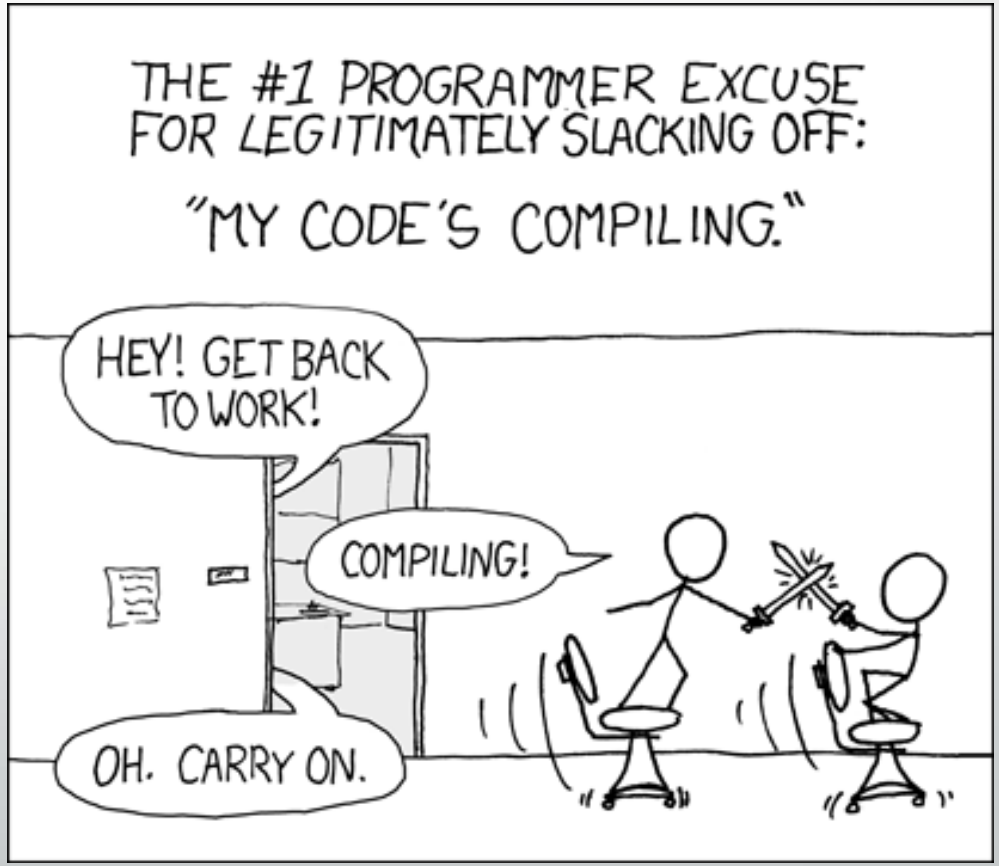
- Configuring not programing
- Diffrence between CPU and FPGA
- Low latency
- But long synthesizing time



# FPGA- field *„programmable”* gate array

- Configuring not programing
- Diffrence between CPU and FPGA
- Low latency
- But long synthesizing time

  (equivalent of compiling)



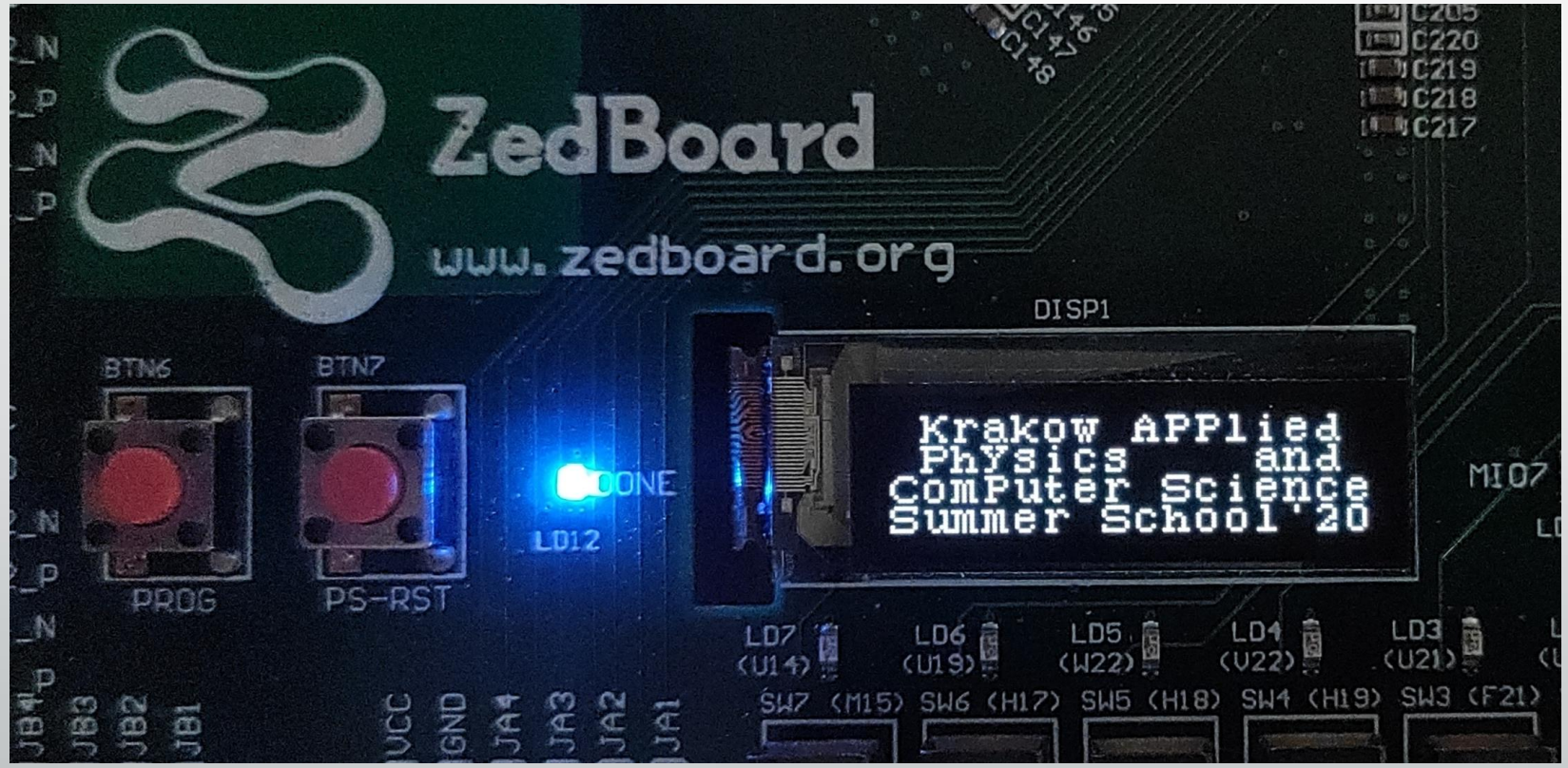
# Verilog

- Hardware description language
- Similar syntax to C

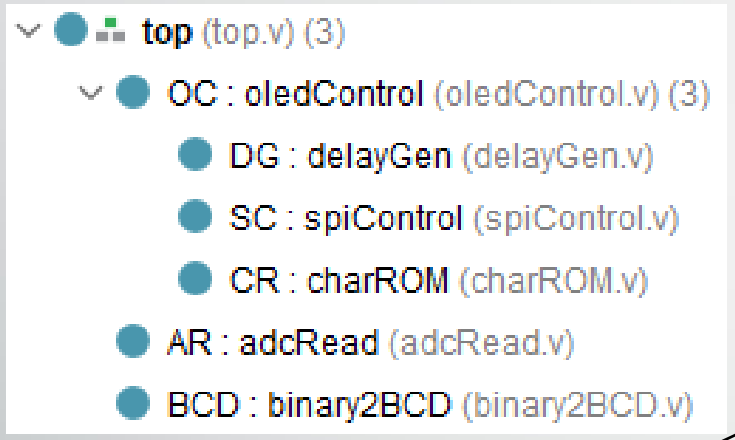
```
module count (input clk, rst,
        output reg [3:0] q);
always @(posedge clk)
        if(rst)
                q <= 4'b0;
        else
                q <= q + 1'b1;
endmodule
```

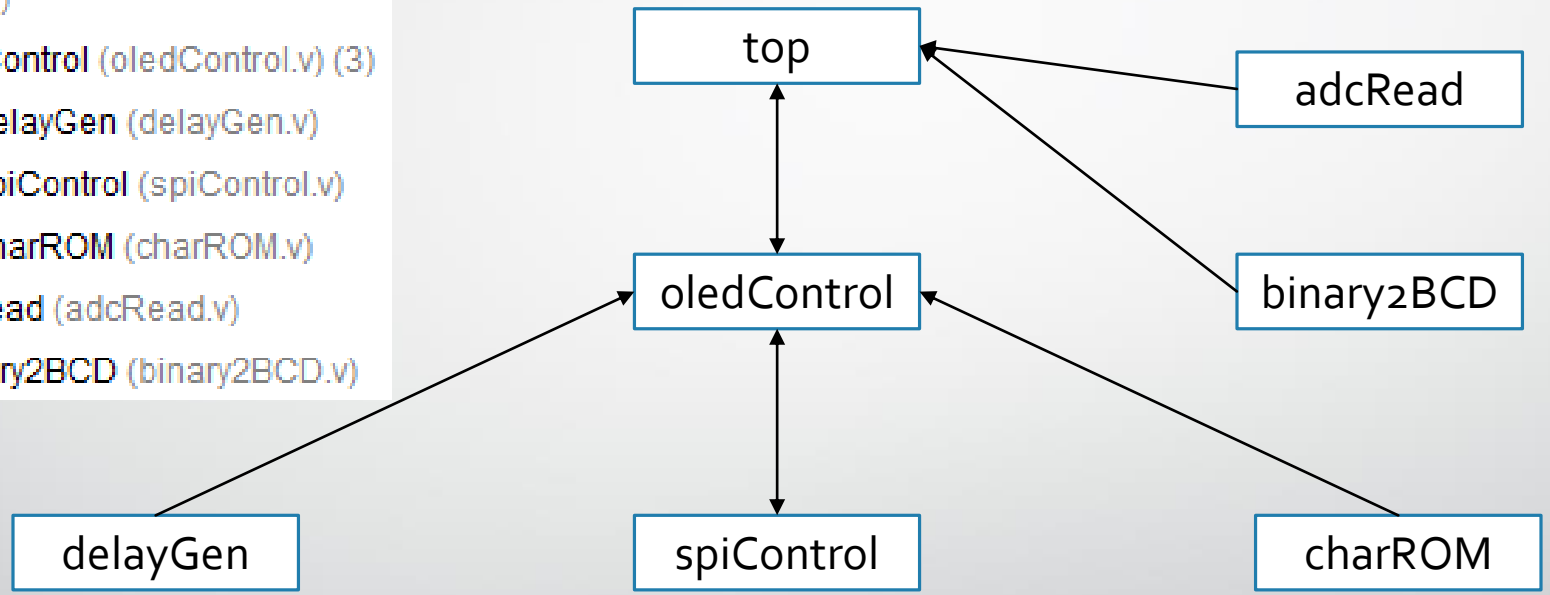
# OLED Display

- ZedBoard Zynq-7000 has built in 128 x 32 OLED display (UG-2832HSWEG04)

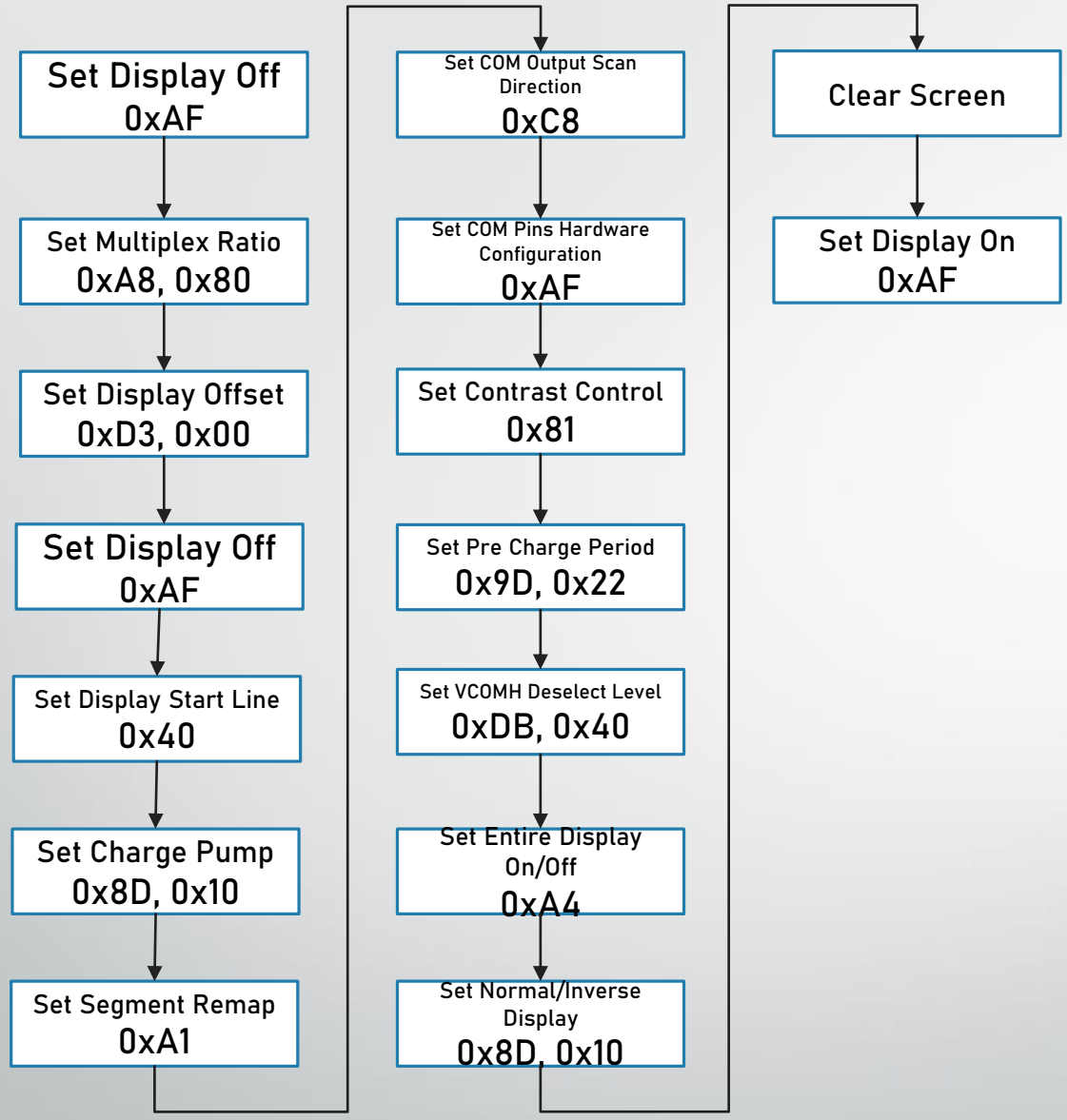


# ADC +OLED Display- project hierarchy



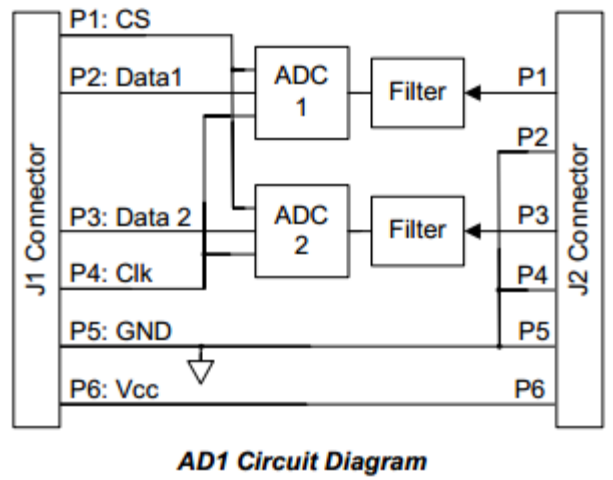
# OLED Initialization




```
IDLE  = 'd0,
DELAY = 'd1,
INIT  = 'd2,
RESET = 'd3,
CHRG_PUMP = 'd4,
CHRG_PUMP1 = 'd5,
WAIT_SPI = 'd6,
PRE_CHRG = 'd7,
PRE_CHRG1 = 'd8,
VBAT_ON = 'd9,
CONTRAST = 'd10,
CONTRAST1 = 'd11,
SEG_REMAP = 'd12,
SCAN_DIR = 'd13,
COM_PIN = 'd14,
COM_PIN1 = 'd15,
DISPLAY_ON = 'd16,
FULL_DISPLAY = 'd17,
DONE = 'd18,
PAGE_ADDR = 'd19,
PAGE_ADDR1 = 'd20,
PAGE_ADDR2 = 'd21,
COLUMN_ADDR = 'd22,
SEND_DATA = 'd23;
```

# ADC- analog to digital converter

- 12 bit digital output
- SPI protocol
- 1 MSPS

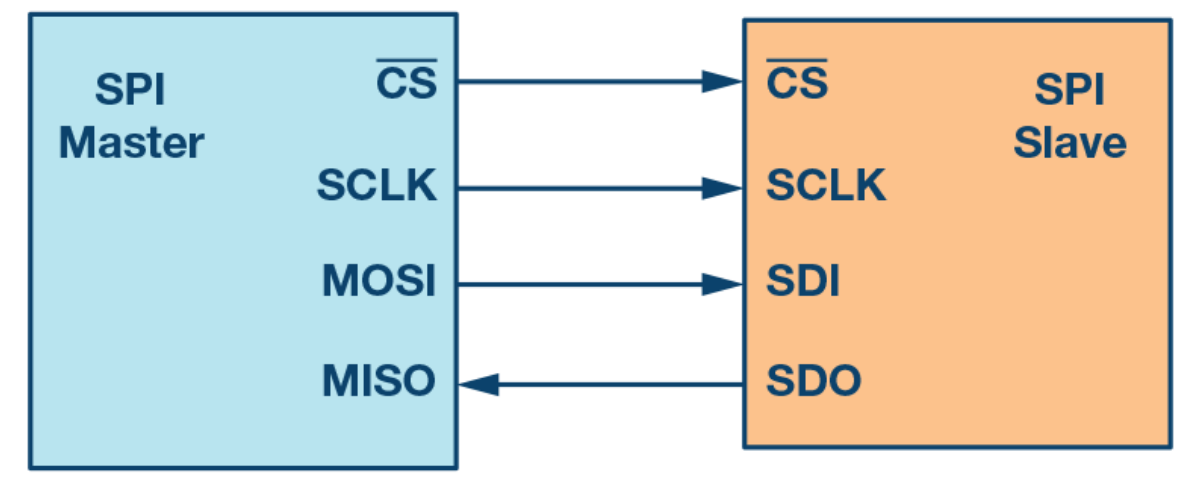



*AD1 Circuit Diagram*

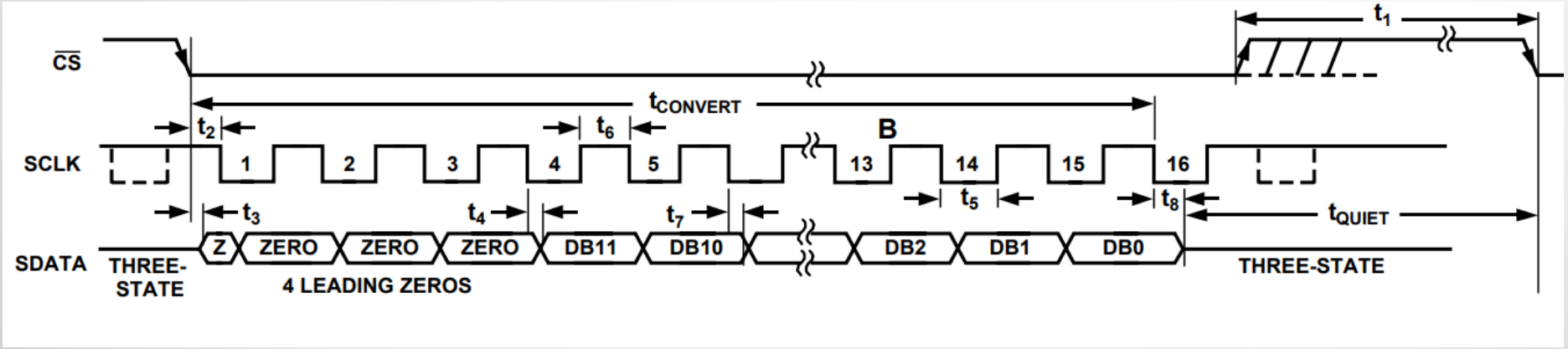
# Serial Peripherial Interface

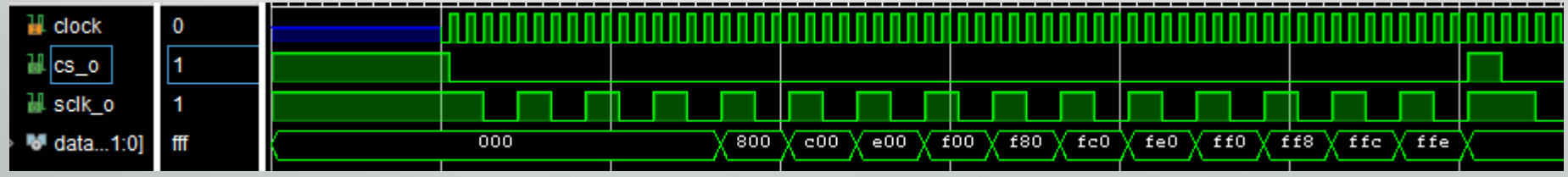
- synchronus serial comunication
- also caled a 4-wire serial bus

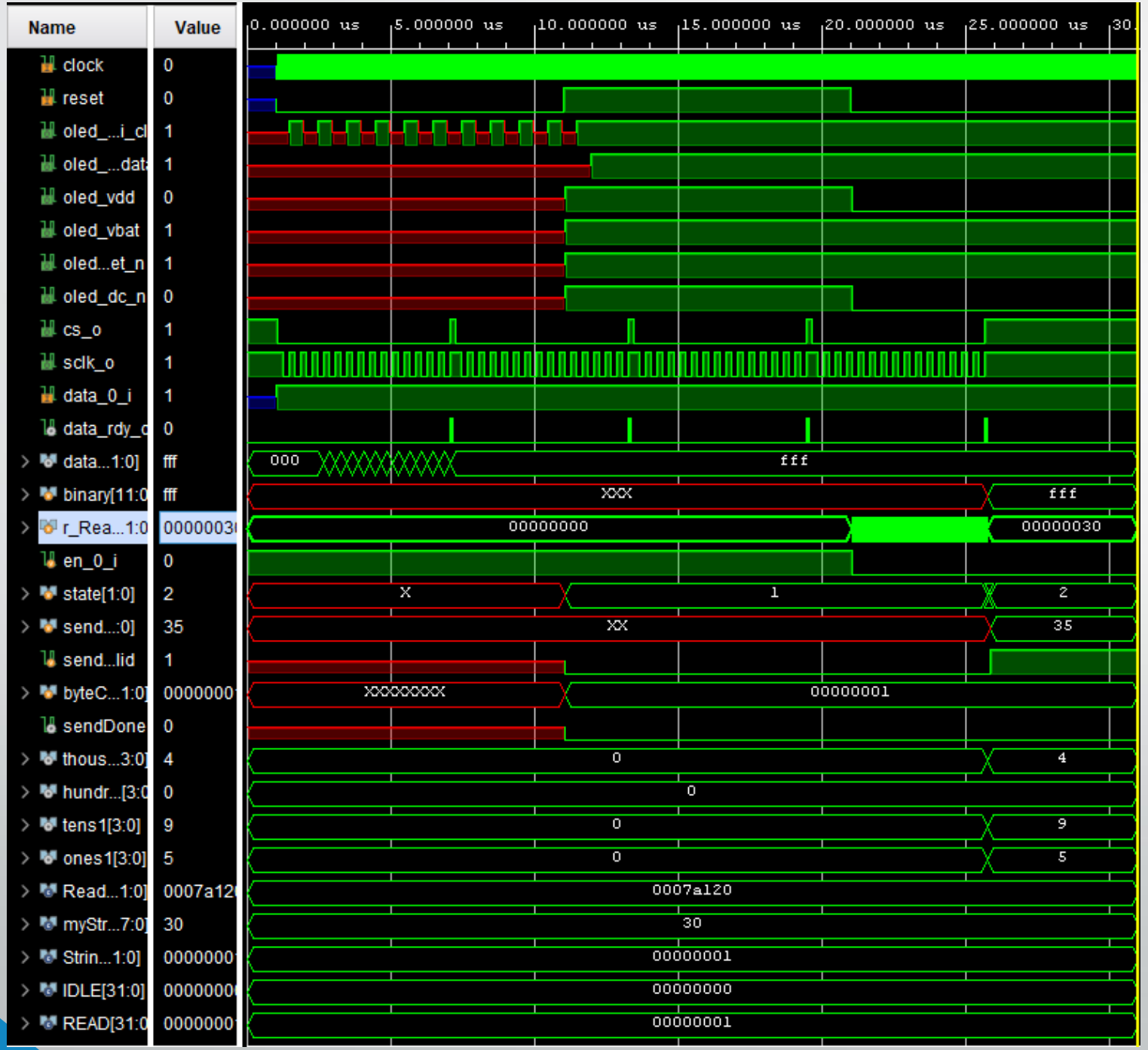
1. Chip select (CS)
2. Clock (SCLK)
3. Master out, slave in (MOSI)
4. Master in, slave out (MISO)

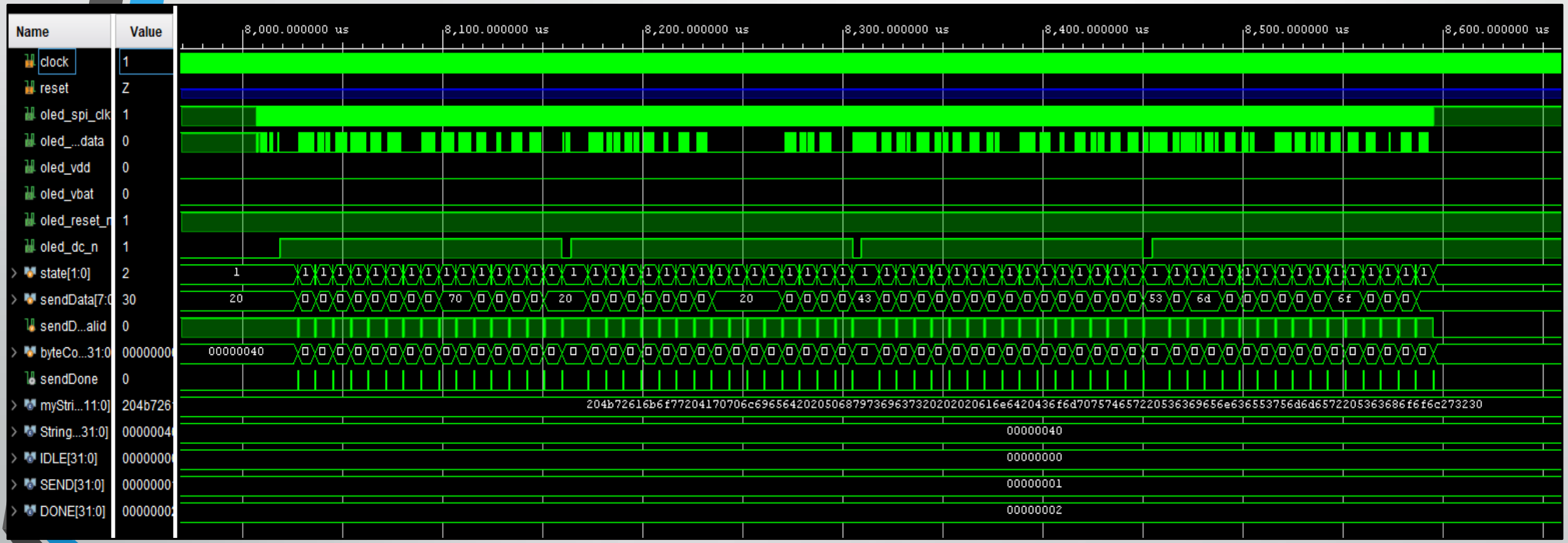


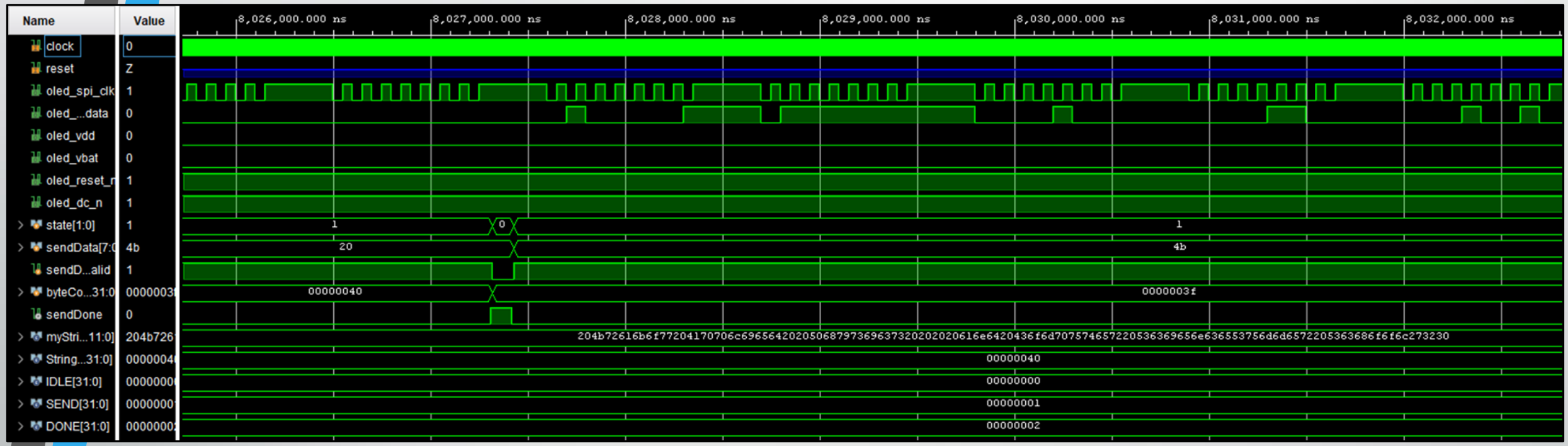
# SPI example on 12 bit ADC




Name
Value
0.000000 us
5.000000 us
10.000000 us
15.000000 us
20.000000 us
25.000000 us
clock 0
reset 0
oled_...i_cl 1
oled_...dat 1
oled_vdd 0
oled_vbat 1
oled...et_n 1
oled_dc_n 0
cs_o 1
sclk_o 1
data_0_i 1
data_rdy_c 0
data...1:0] fff
binary[11:0 fff
r_Rea...1:0 0000003
en_0_i 0
state[1:0] 2
send...:0] 35
send...lid 1
byteC...1:0] 0000000
sendDone 0
thous...3:0] 4
hundr...[3:0 0
tens1[3:0] 9
ones1[3:0] 5
Read...1:0] 0007a12
myStr...7:0] 30
Strin...1:0] 0000000
IDLE[31:0] 0000000
READ[31:0 0000000
000
fff
XXX
fff
00000000
00000030
X
1
2
XX
35
XXXXXXXX
00000001
0
4
0
0
9
0
5
0007a120
30
00000001
00000000
00000001


Name
Value
8,000.000000 us
8,100.000000 us
8,200.000000 us
8,300.000000 us
8,400.000000 us
8,500.000000 us
8,600.000000 us
clock
1
reset
Z
oled_spi_clk
1
oled_...data
0
oled_vdd
0
oled_vbat
0
oled_reset_n
1
oled_dc_n
1
state[1:0]
2
sendData[7:0]
30
sendD...alid
0
byteCo...31:0]
00000000
sendDone
0
myStri...11:0]
204b726
String...31:0]
0000004
IDLE[31:0]
0000000
SEND[31:0]
0000000
DONE[31:0]
0000000
204b72616b6f77204170706c6965642020506879736963732020202020616e6420436f6d70757465722053636369656e636553756d6d6572205363686f6f6c273230
00000040
00000000
00000001
00000002


Name
Value
8,026,000.000 ns
8,027,000.000 ns
8,028,000.000 ns
8,029,000.000 ns
8,030,000.000 ns
8,031,000.000 ns
8,032,000.000 ns
clock
0
reset
Z
oled_spi_clk
1
oled_...data
0
oled_vdd
0
oled_vbat
0
oled_reset_n
1
oled_dc_n
1
state[1:0]
1
1
0
1
sendData[7:0]
4b
20
4b
sendD...alid
1
byteCo...31:0]
0000003f
00000040
0000003f
sendDone
0
myStri...11:0]
204b726f
204b72616b6f77204170706c6965642020506879736963732020202061 6e6420436f6d70757465722053636965 6e636553756d6d6572205363686f6f6c273230
String...31:0]
00000040
00000040
IDLE[31:0]
00000000
00000000
SEND[31:0]
00000001
00000001
DONE[31:0]
00000002
00000002

# Thank you!