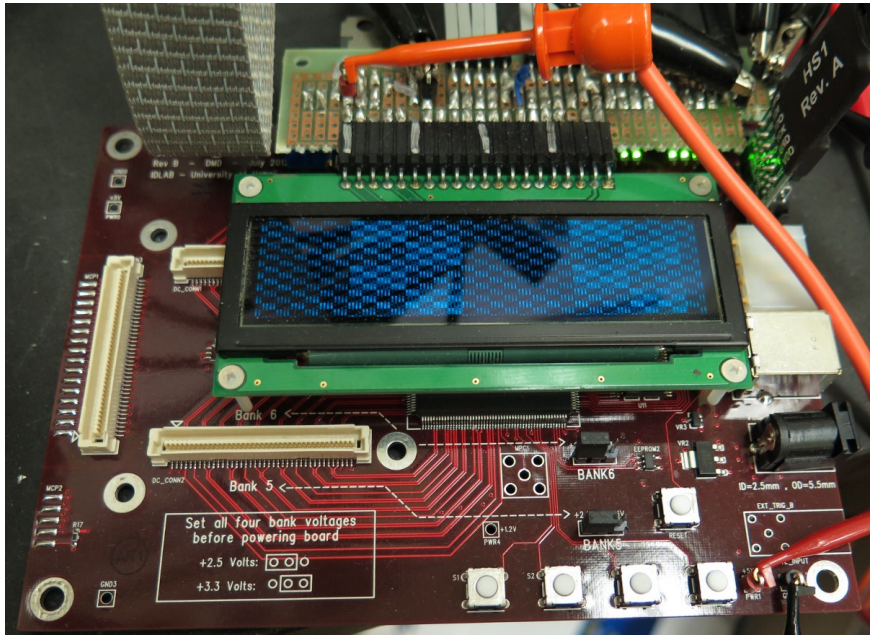
example project to interface to M6800-style peripheral

- OLED display control project:
  - https://code.google.com/p/idlab-general/source/browse/universal_eval/OLED_display/
- clock_enable_generator:
  - http://code.google.com/p/idlab-scrod/source/browse/SCROD-boardstack/new_daq_interface/src/utilities/clock_enable_generator.vhd



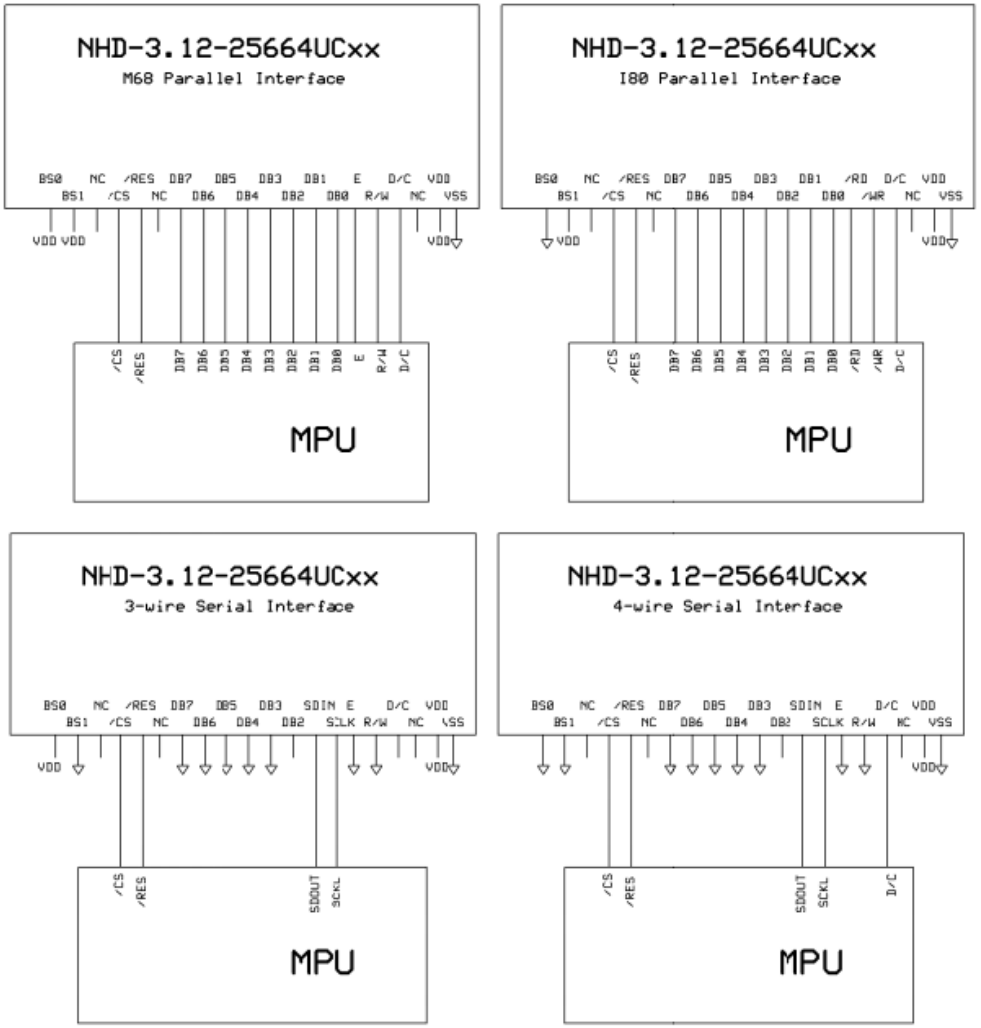
example interface:
OLED display (M6800)

- page 5 of datasheet:
  - shows four interface options
- page 10:
  - shows signal descriptions

MPU Interface Pin Assignment Summery

Bus Interface	Data/Command Interface								Control Signals				
	D7	D6	D5	D4	D3	D2	D1	D0	E	R/W	/CS	D/C	/RES
8-bit 6800	D[7:0]								E	R/W	/CS	D/C	/RES
8-bit 8080	D[7:0]								/RD	/WR	/CS	D/C	/RES
3-wire SPI	Tie LOW					NC	SDIN	SCLK	Tie LOW		/CS	Tie LOW	/RES
4-wire SPI	Tie LOW					NC	SDIN	SCLK	Tie LOW		/CS	D/C	/RES

Wiring Diagrams



6800-MPU Parallel Interface

The parallel interface consists of 8 bi-directional data pins, R/W, D/C, E, and /CS.
A LOW on R/W indicates write operation, and HIGH on R/W indicates read operation.
A LOW on D/C indicates "Command" read or write, and HIGH on D/C indicates "Data" read or write.
The E input serves as data latch signal, while /CS is LOW. Data is latched at the falling edge of E signal.

Function	E	R/W	/CS	D/C
Write Command	↓	0	0	0
Read Status	↓	1	0	0
Write Data	↓	0	0	1
Read Data	↓	1	0	1

OLED display timing

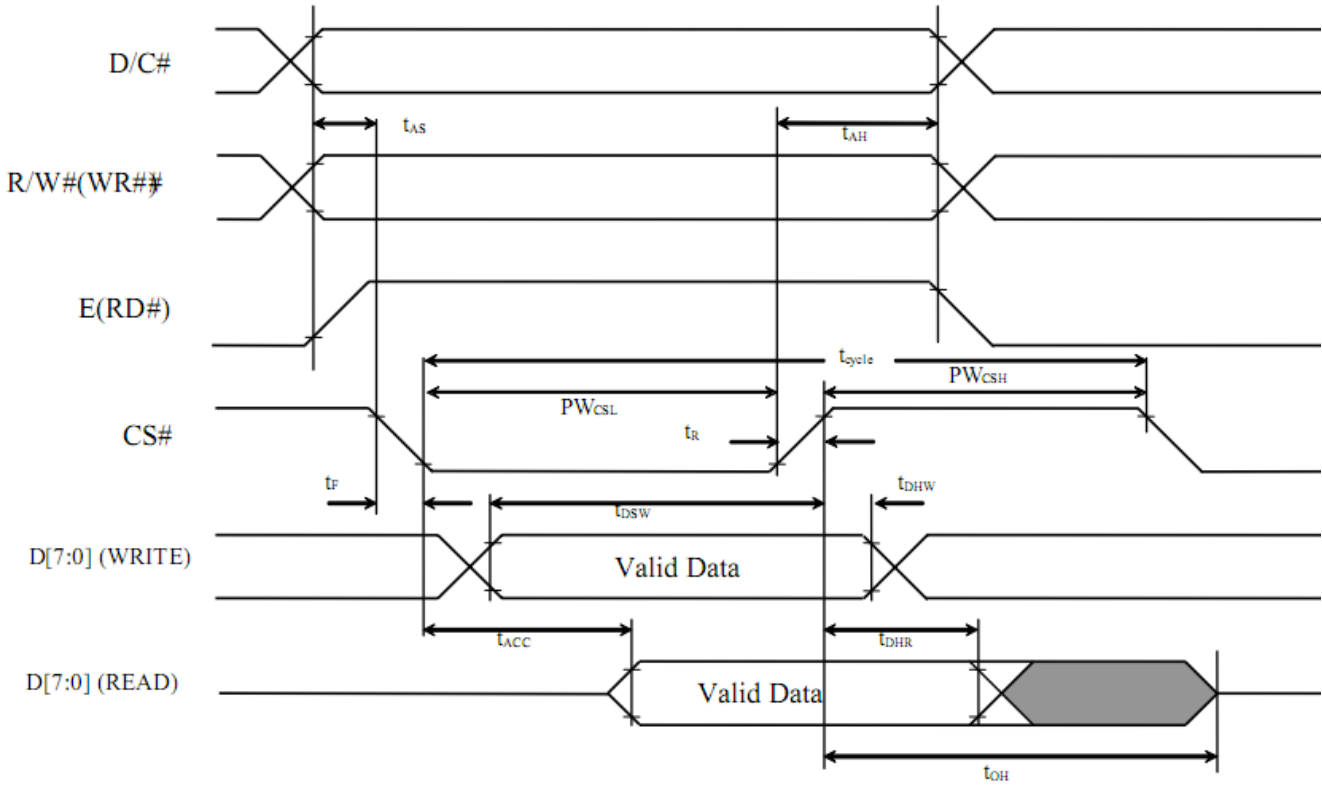
- page 51 of display controller's datasheet

Table 13-2 : 6800-Series MCU Parallel Interface Timing Characteristics

(V_{DD} - V_{SS} = 2.4 to 2.6V, V_{DDIO}=1.6V, V_{CI} = 3.3V, T_A = 25°C)

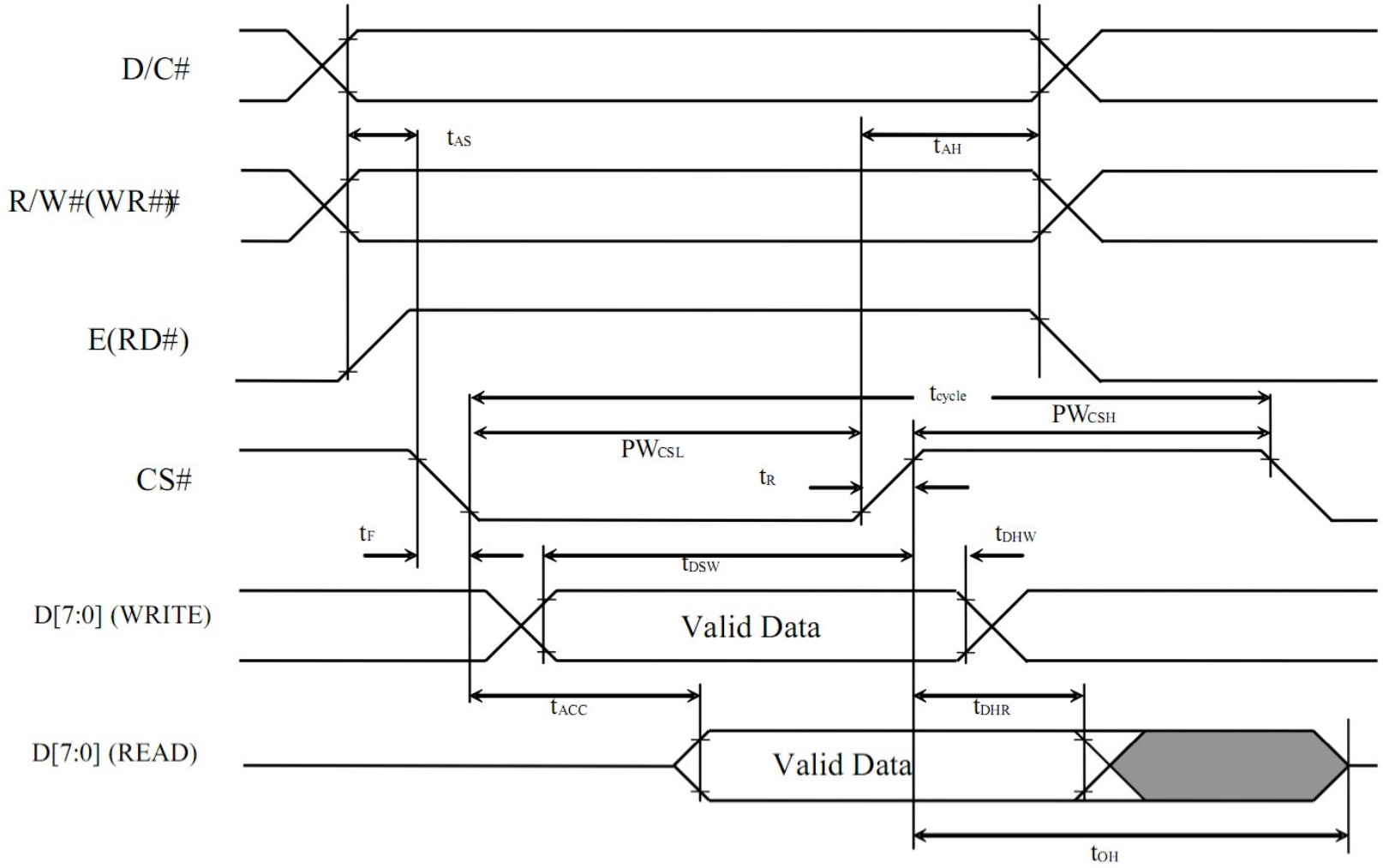
Symbol	Parameter	Min	Typ	Max	Unit
t_{cycle}	Clock Cycle Time	300	-	-	ns
t_{AS}	Address Setup Time	10	-	-	ns
t_{AH}	Address Hold Time	0	-	-	ns
t_{DSW}	Write Data Setup Time	40	-	-	ns
t_{DHW}	Write Data Hold Time	7	-	-	ns
t_{DHR}	Read Data Hold Time	20	-	-	ns
t_{OH}	Output Disable Time	-	-	70	ns
t_{ACC}	Access Time	-	-	140	ns
PW_{CSL}	Chip Select Low Pulse Width (read) Chip Select Low Pulse Width (write)	120 60	-	-	ns
PW_{CSH}	Chip Select High Pulse Width (read) Chip Select High Pulse Width (write)	60 60	-	-	ns
t_R	Rise Time	-	-	15	ns
t_F	Fall Time	-	-	15	ns

Figure 13-1 : 6800-series MCU parallel interface characteristics



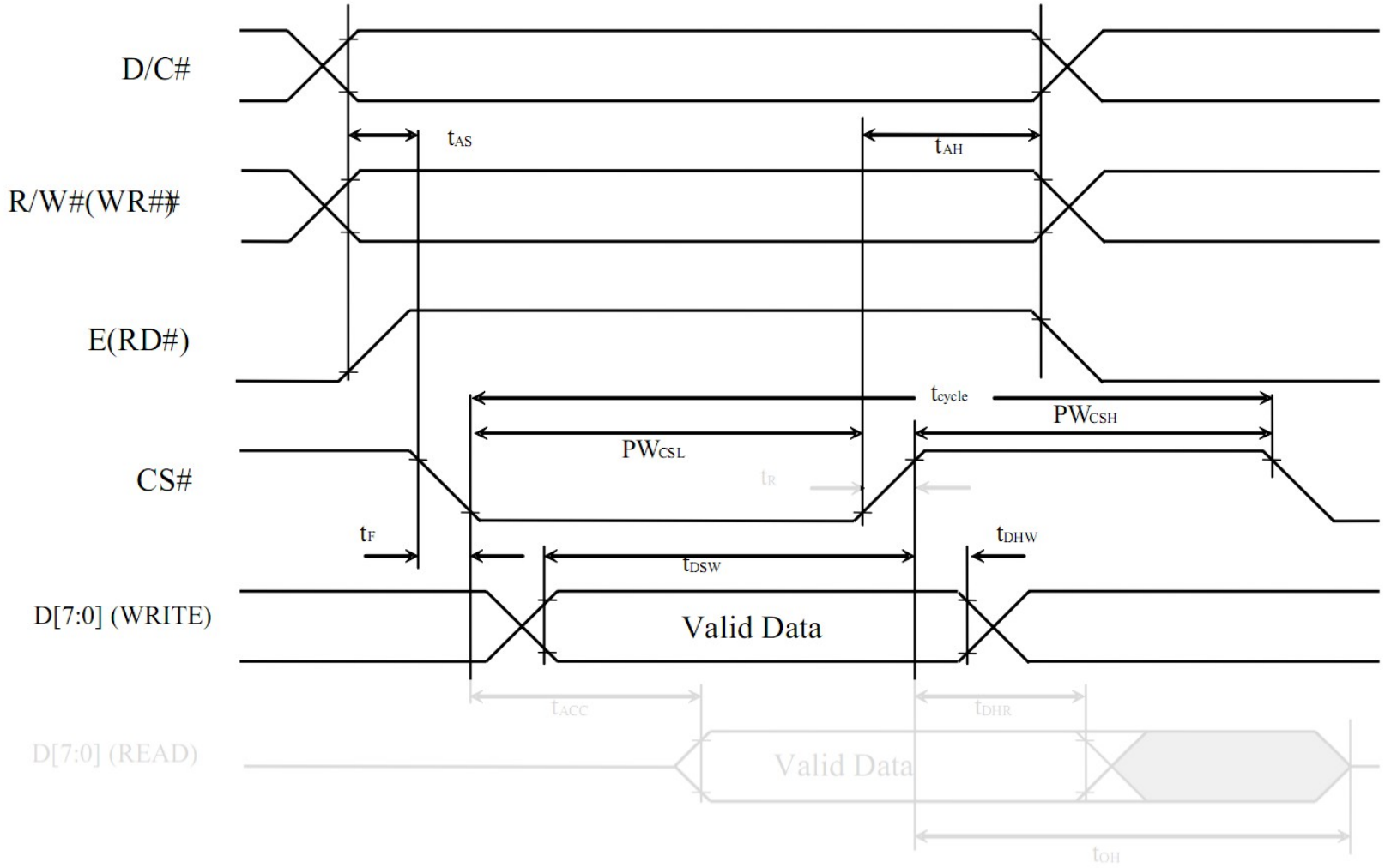
OLED timing diagram (original)

Figure 13-1 : 6800-series MCU parallel interface characteristics



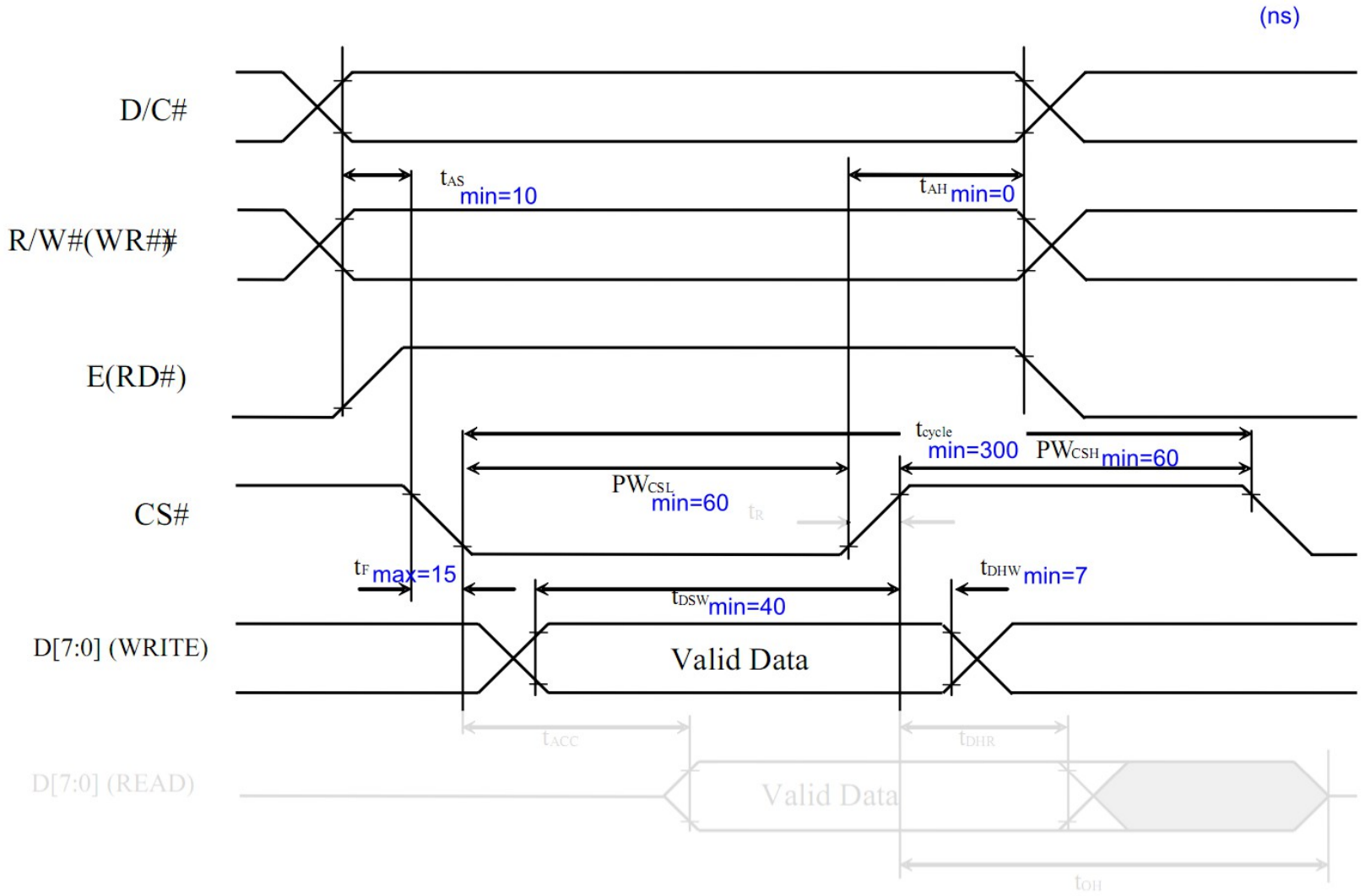
OLED timing diagram (ignore irrelevant parts)

Figure 13-1 : 6800-series MCU parallel interface characteristics



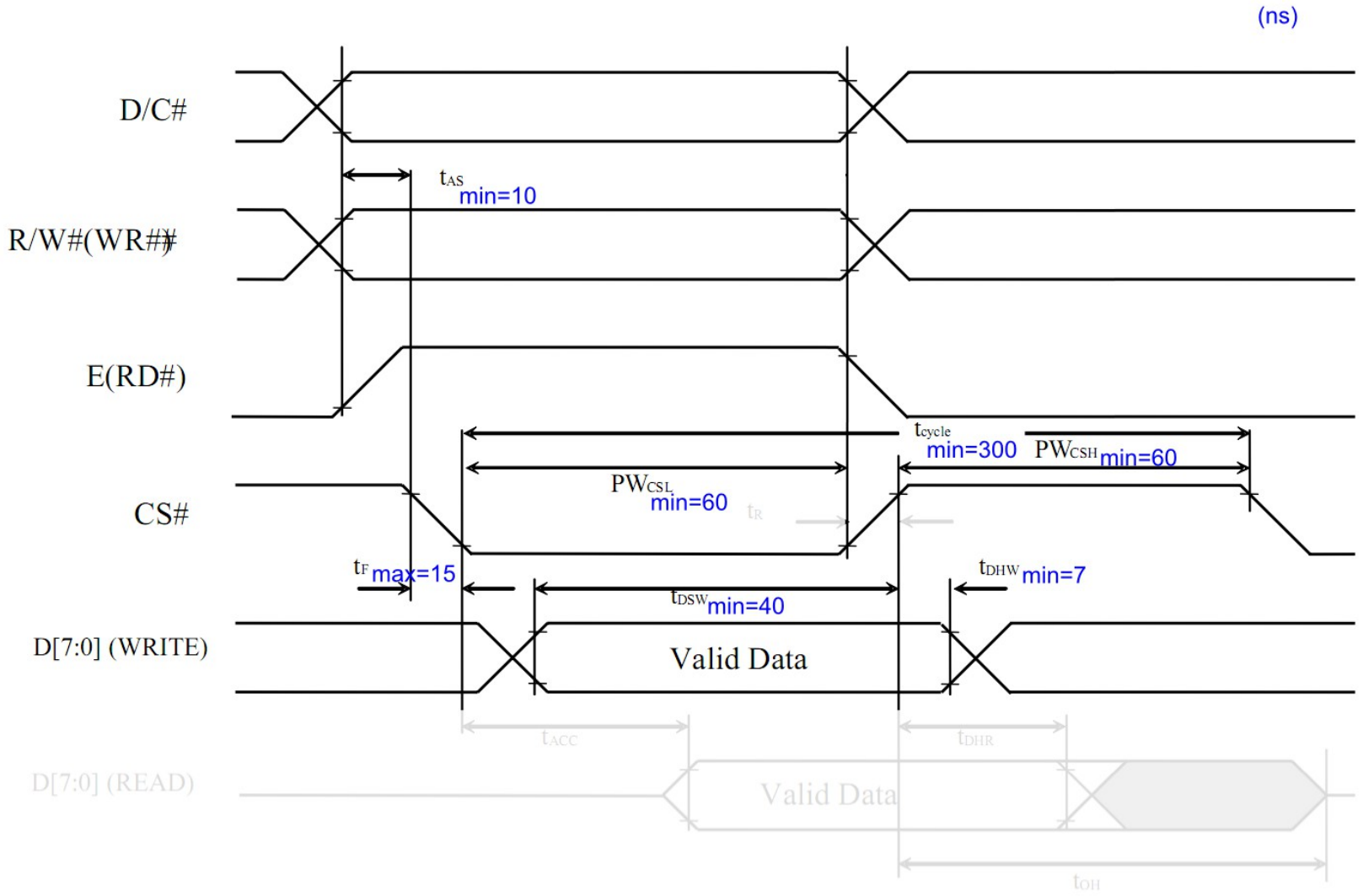
OLED timing diagram (annotate with min/max constraints)

Figure 13-1 : 6800-series MCU parallel interface characteristics



OLED timing diagram (make adjustments to simplify)

Figure 13-1 : 6800-series MCU parallel interface characteristics

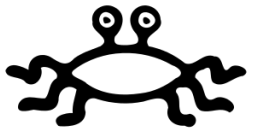


OLED display control loop

val\OLED_display\ise-project\OLED_display.xise - [OLED_display.vhdl]

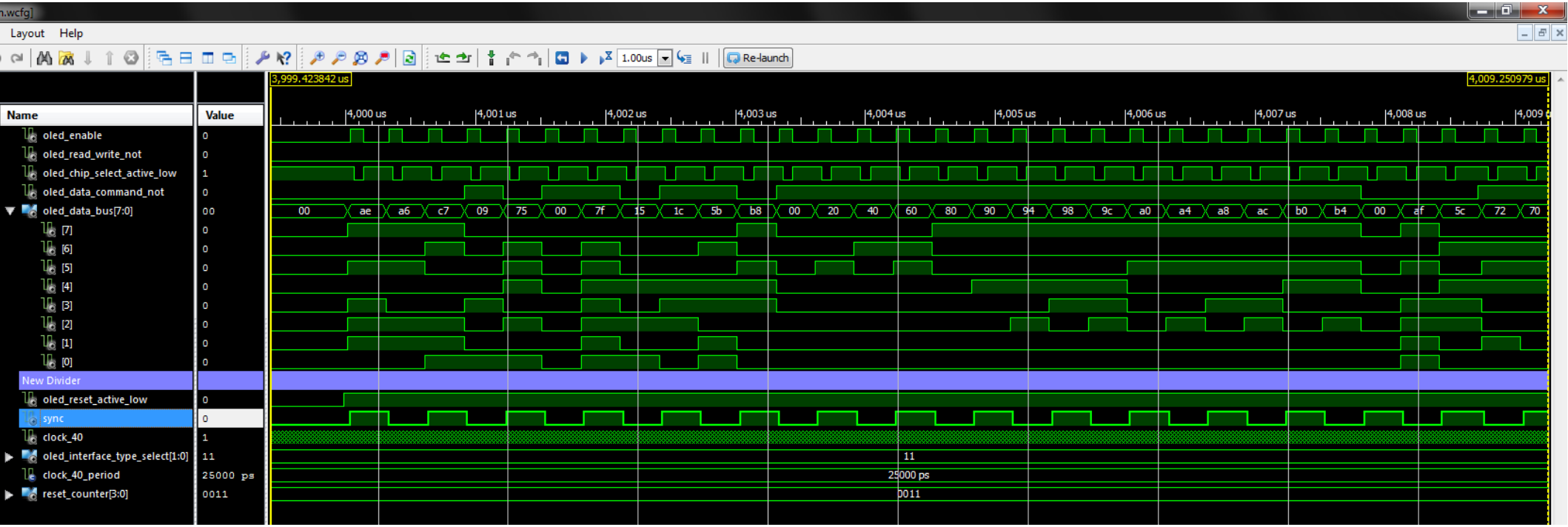
t Help

```
if (transaction_required = '1') then
   if (individual_transaction_counter < 10) then      -- total 300 ns cycle time
                                                      -- (takes a cycle to get here from elsewhere)
      transaction_in_progress <= '1';
      if (individual_transaction_counter < 1) then    -- itc = 0 (for 25 ns)
         internal_sync <= not internal_sync;
         internal_enable <= '1';
      elsif (individual_transaction_counter < 4) then -- itc = 1, 2, 3 (for 75 ns)
         internal_chip_select <= '1';
      else                                            -- itc = 4, 5, 6, 7, 8, 9 (for 150 ns)
         internal_chip_select <= '0';
         internal_enable <= '0';
      end if;
      individual_transaction_counter <= individual_transaction_counter + 1;
   else                                               -- itc = 10 (for 25 ns)
      transaction_required <= '0';
      transaction_in_progress <= '0';
      individual_transaction_counter <= (others => '0');
   end if;
end if;
```

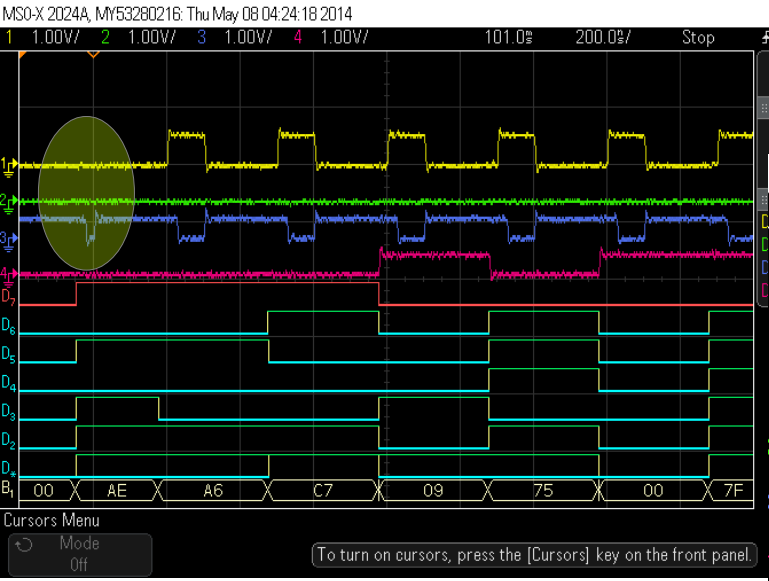


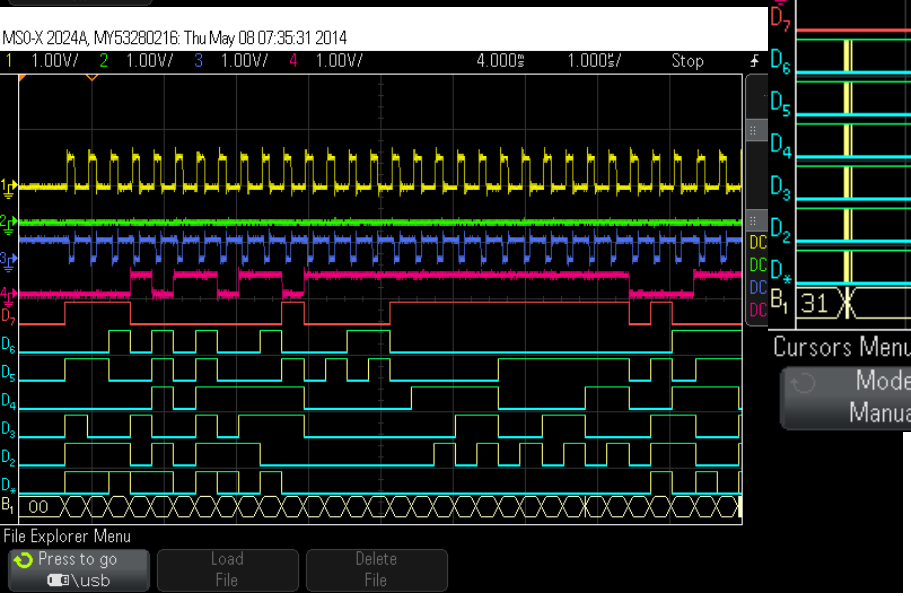
mary (Implemented) | OLED_display.vhdl

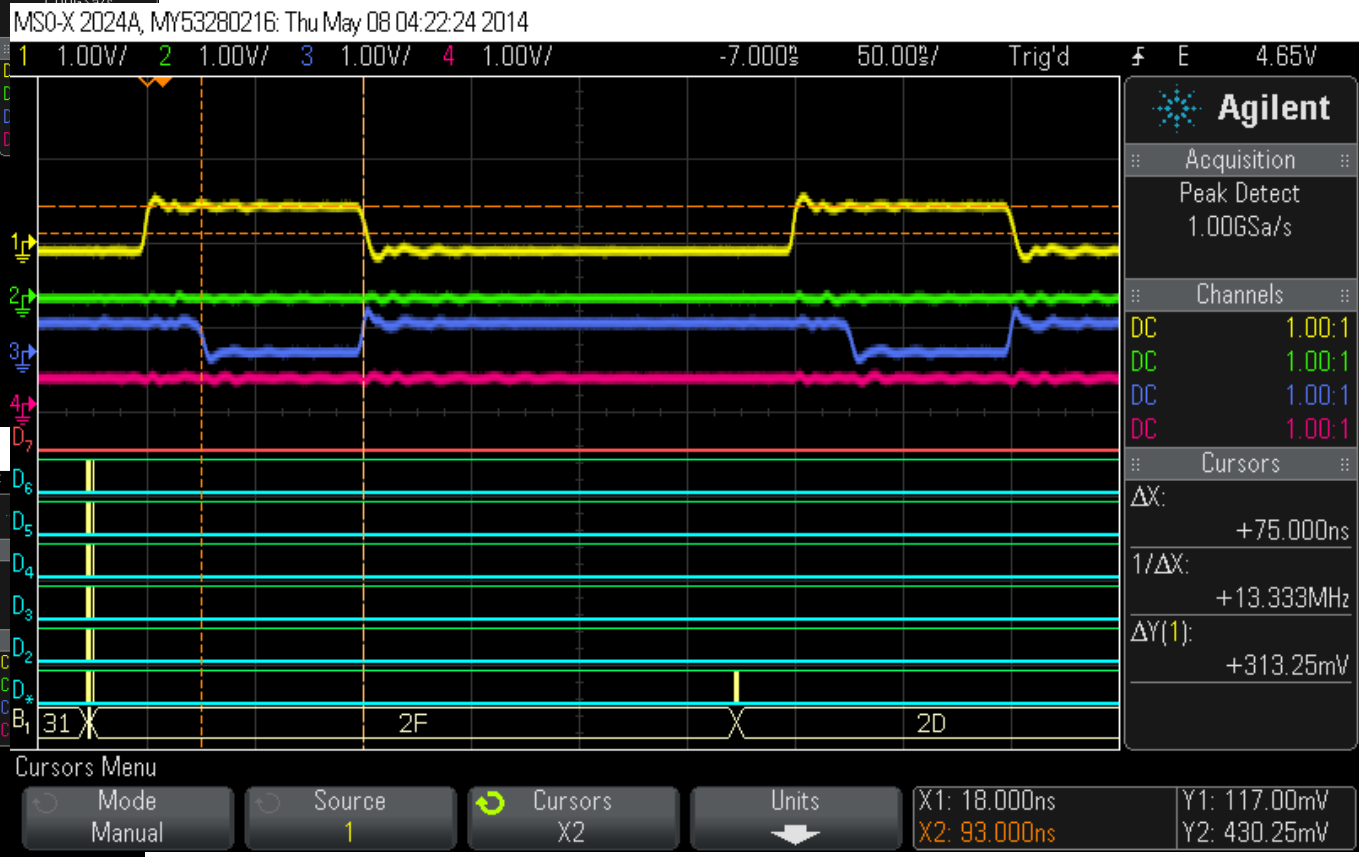
testbench / simulation



scope traces

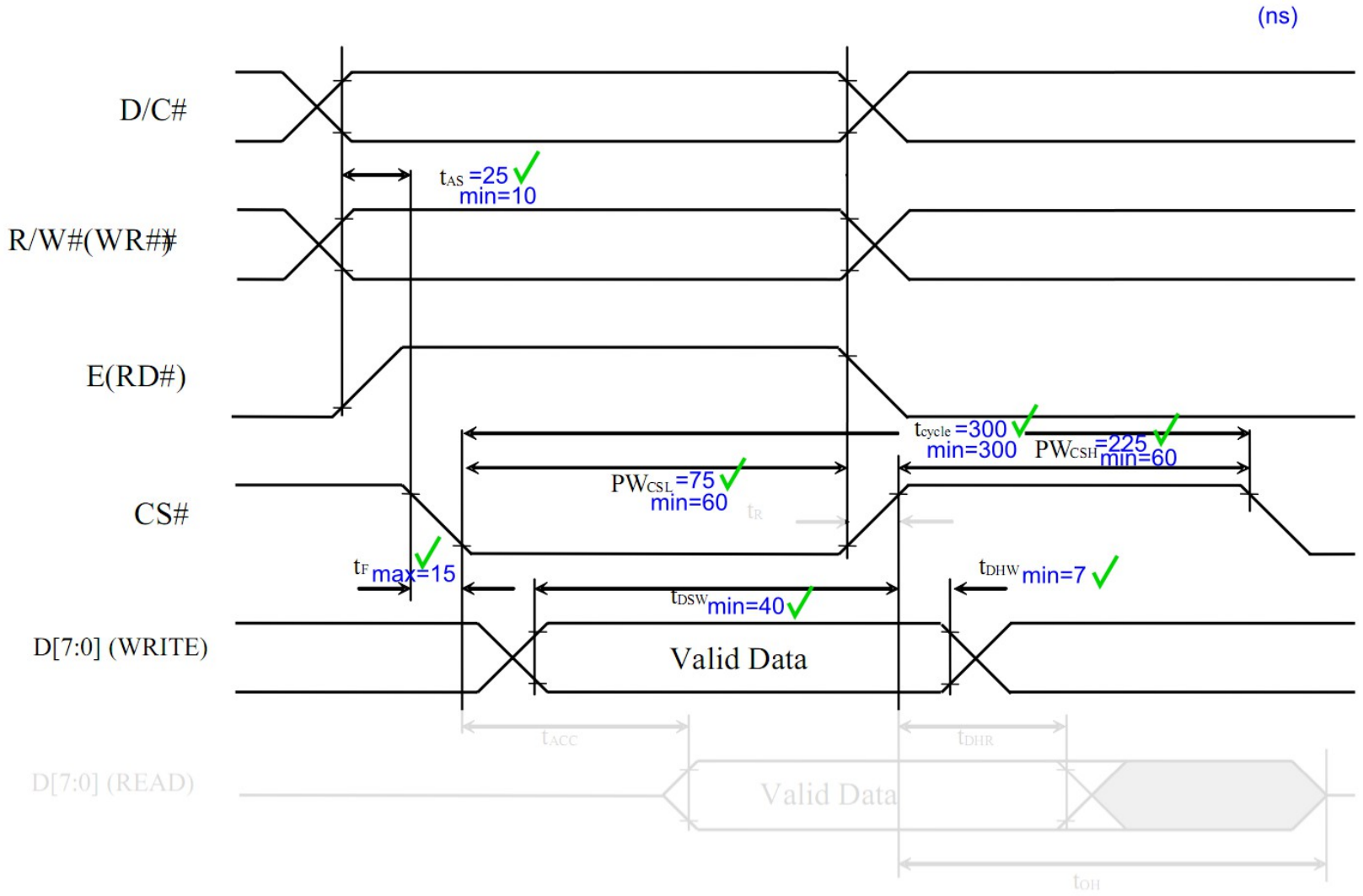






OLED timing diagram (verify met all requirements)

Figure 13-1 : 6800-series MCU parallel interface characteristics



advice

```
entity Xample is
   generic (
      constant mywidth : integer := 8
   );
   port (
      clock    : in    std_logic;
      data_bus : inout std_logic_vector(mywidth-1 downto 0);
      parity   :   out std_logic
```

- use constants where possible:
  - to reduce instances of “magic numbers:”
    - constant mywidth : integer := integer(ceil(log2(1300.0)));
  - for future expandability:
    - signal mycounter : unsigned(mywidth-1 downto 0);
- better yet, use generics!
- don't be intimidated by the flurry of messages that go by as it compiles
  - pay attention to the warnings that are *your* fault; these are clues as to what you are doing wrong

```
pgen : entity work.Xample
   generic map (
      mywidth => db_width
   )
   port map (
      clock    => clk,
      data_bus => data,
      parity   => par
```

warning messages

```
**************************
Generating Clock Report
**************************

+---------------------+-------------+------+------+------------+-------------+
|        Clock Net    |   Resource  |Locked|Fanout|Net Skew(ns)|Max Delay(ns)|
+---------------------+-------------+------+------+------------+-------------+
|     clock_150_BUFGP |      BUFGMUX5| No   |   62 |  0.019     |  1.033      |
+---------------------+-------------+------+------+------------+-------------+

* Net Skew is the difference between the minimum and maximum routing
only delays for the net. Note this is different from Clock Skew which
is reported in TRCE timing report. Clock Skew is the difference between
the minimum and maximum path delays which includes logic delays.

* The fanout is the number of component pins not the individual BEL loads,
for example SLICE loads not FF loads.

Timing Score: 304773 (Setup: 304773, Hold: 0, Component Switching Limit: 0)

WARNING:Par:468 - Your design did not meet timing.  The following are some suggestions to assist you to
meet timing in
   your design.

   Review the timing report using Timing Analyzer (In ISE select "Post-Place &
   Route Static Timing Report"). Go to the failing constraint(s) and evaluate the failing paths for each
constraint.

   Try the Design Goal and Strategies for Timing Performance(In ISE select Project -> Design Goals &
Strategies) to
   ensure the best options are set in the tools for timing closure.

   Use the Xilinx "SmartXplorer" script to try special combinations of
   options known to produce very good results.

   Visit the Xilinx technical support web at http://support.xilinx.com and go to
   either "Troubleshoot->Tech Tips->Timing & Constraints" or "
   TechXclusives->Timing Closure" for tips and suggestions for meeting timing
   in your design.

Asterisk (*) preceding a constraint indicates it was not met.
   This may be due to a setup or hold violation.

----------------------------------------------------------------------------------------------------------
  Constraint                                |    Check    | Worst Case |  Best Case | Timing |   Timing
                                            |             |    Slack   | Achievable | Errors |    Score
----------------------------------------------------------------------------------------------------------
* ts_clock_150 = PERIOD TIMEGRP "clock_150" | SETUP       |    -7.488ns|    14.155ns|     106|
304773
   6.667 ns HIGH 50%                        | HOLD        |     0.891ns|            |       0|
0
----------------------------------------------------------------------------------------------------------


1 constraint not met.


Generating Pad Report.

All signals are completely routed.

Total REAL time to PAR completion: 15 secs
```

```
**************************
Generating Clock Report
**************************

+---------------------+-------------+------+------+------------+-------------+
|        Clock Net    |   Resource  |Locked|Fanout|Net Skew(ns)|Max Delay(ns)|
+---------------------+-------------+------+------+------------+-------------+
|     clock_40_BUFGP |      BUFGMUX0| No   |   71 |  0.020     |  1.034      |
+---------------------+-------------+------+------+------------+-------------+

* Net Skew is the difference between the minimum and maximum routing
only delays for the net. Note this is different from Clock Skew which
is reported in TRCE timing report. Clock Skew is the difference between
the minimum and maximum path delays which includes logic delays.

* The fanout is the number of component pins not the individual BEL loads,
for example SLICE loads not FF loads.

Timing Score: 0 (Setup: 0, Hold: 0, Component Switching Limit: 0)

Asterisk (*) preceding a constraint indicates it was not met.
   This may be due to a setup or hold violation.

----------------------------------------------------------------------------------------------------------
  Constraint                                |    Check    | Worst Case |  Best Case | Timing |   Timin
                                            |             |    Slack   | Achievable | Errors |    Scor
----------------------------------------------------------------------------------------------------------
  ts_clock_40 = PERIOD TIMEGRP "clock_40" 2 | SETUP       |     8.394ns|    16.606ns|       0|
0
  5 ns HIGH 50%                             | HOLD        |     0.848ns|            |       0|
0
----------------------------------------------------------------------------------------------------------


All constraints were met.


Generating Pad Report.

All signals are completely routed.

WARNING:Par:283 - There are 1 loadless signals in this design. This design will cause Bitgen to issue
DRC warnings.
```

more advice

- don't use up all the global clocks; use a single clock and distribute clock enables with the appropriate duty cycles
- read through coding recommendations from Xilinx (and follow them!):
  - http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_7/sim.pdf
  - http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_7/xst_v6s6.pdf
- if possible, use someone else's *working* code
- build things one at a time and ***test them***!

parting words...

- A complex system that works is invariably found to have evolved from a simple system that worked. A complex system designed from scratch never works and cannot be patched up to make it work. You have to start over with a working simple system. – John Gall (1975)