**-\***

Software

Requirements Specification

For

4 x Gen4L-block Tester software

Version 01

|  |  |  |
| --- | --- | --- |
|  | **Name** | **Date** |
| Written by: | Eldad Chen |  |
| Approved by: |  |  |
| Version No: | Version 01 |  |
|  |  |  |

1. **Overview**

The Gen4L tester program will allow a comprehensive testing of the Gen4L-based print-block.. The hardware configuration of the tester will be based on the electronics designed for the printer, namely, OHDB 2 + GEN4L Head driver Board. Some changes to the OHDB firmware may be needed in order to support the tester functionality.

The tester program can be based on the existing Eden Tester software which will undergo changes emanating from the requirements presented in this document.

1. **Tester functions**
   1. **Temperature control**
      1. Set temperature for 4 ptintheads, each containing ODD and EVEN heaters (8 heaters altogether); Set temperature for 4 channel block heaters (pre-heater1/2, block front/rear).

Setting temperature for heads and block is done in Main MCU. The Main MCU responsible for operating the heaters (by writing to a register in the FPGA) and reads the temperature sensors. By doing that it can control the heating loop.

* + 1. Actual printheads and block temperature reading display (8+4 channels)

The temperature sensors are read through A2D MCU.

At E1 heads configuration, we read every head’s temprature through pin 17 of the 30pin head’s connector. HD1-HD8 readings go to A2D MCU as AN1-AN8 to P3.

Block heaters go to A2D MCU as AN9-AN12 to P1(0)-P1(3) respectively.

Block pre-heater is read by A2D MCU as AN22 at P0(5).

At GEN4 configuration, we are using pin 23 (that was a spare/unused pin) as the second head temperature reading.

* **Remark:** switching from E1 to GEN4 configuration needs resistors change in OHDB2 board.

There are two options to configure GEN4 heads: insert head drives boards in HD1,3,5,7 location or in HD2,4,6,8 location. Every configuration needs different hardware assembly. In the two configurations the analog heads temprature readings order remain the same as E1.

All block heaters are the same as in E1 configuration.

* + 1. Set error margin

Main MCU parameter.

* + 1. Heater ON/OFF control (global)

There is some mechanism for operating the heaters.

One part of it is enabling all heaters and after approximate 2 seconds the FPGA disables them automatically.

This part is controlled by writing (the value isn’t matter) to register WR\_HEAT1 at address 06h, to enable the heaters for up to 2 seconds.

Second part is enabling or disabling continuously the heaters. The software is the only one that can change this state.

Writing ‘1‘ to register WR\_DISEN at address 68h will enable the heaters.

Writing ‘0’ will disable all heaters.

* + 1. Individual Enable / Disable control for each heater.

All heaters (block and heads) are controlled from the FPGA.

Register name: WR\_HTRS

Heaters register address: 60h

Bits D0-D7 operate HD1-HD8 respectively

Bits D8-D11 operate block heater1-heater4 respectively.

Bit D12 operates block pre-heater

* + 1. Temperature-in-range / Temperature-out-of-range indication.

Set and controlled in Main MCU.

* + 1. Temperature ramp-up control.

Section 7.3 “HEAT UP PROCESS” at GEN4L head integration guide Rev E document describes the limitation: “the power to the heaters should be regulated to less than

0.5C per second at thermistor location”

* 1. **Data setting**
     1. Individual nozzle setting for each of the 4 printheads (4 x 384 nozzles).

Defined and set by GUI.

* + 1. Set ALL/Clear ALL control for all printheads.

Defined and set by GUI.

* + 1. Set/Clear data for a group of nozzles (GUI dependent)

Defined and set by GUI.

* + 1. Load/Save pattern capability.

Defined and set by GUI.

* + 1. Cyclic data generation (cyclic ON/OFF pattern).
  1. **Strobe LED control**
     1. Setting strobe pulse width

Strobe pulse width will be defined by register STRB\_PLS\_WIDTH at address TBD.

Duration of pulse width will be between 0 – 10usec.

* + 1. Setting strobe pulse delay (from fire pulse) –static

See section 2.3.3

* + 1. Setting strobe pulse delay – dynamic (start delay, stop delay, increment, rate(increment delay))

Time delay between fire pulse (first pulse) and strobe pulse will be defined by several registers:

MIN\_STRB\_DLY – setting the minimum time delay (start delay time) between fire pulse and strobe pulse. Range can be 0 – 10usec. Register address TBD

MAX\_STRB\_DLY – setting maximum delay time between the two pulses

defined by register FIRE2STRB\_DLY register at address TBD.

Duration of this delay will be between 0 – 200usec.

STRB\_INC\_STP – This register defines the delay increment step. Every step is a multiplication of 1usec. If register value=00h, the delay will be static and its value will be the difference between MAX\_STRB\_DLY and MIN\_STRB\_DLY.

Register address – TBD

STRB\_INC\_DLY – defines the delay between each step incremental. Range 0-1msec

Register address - TBD

* + 1. LED ON/OFF control

Register WR\_STROBE\_EN, address TBD

D0=’0’ – disable.

D0=’1’ – enable.

* 1. **Fire control**
     1. Fire frequency setting

Register WR\_SIMFR at address 35h control the fire frequency.

The formula is: 33333333/(16\*Frequency)

* + 1. Pulse width setting

TBD

* + 1. Fire Mode setting
       1. Continuous

Putting state machine in continues mode will be done by writing ‘1’ to bit TBD in WR\_DIAG2 register (address TBD).

* + - 1. Duty cycle (# of fires ON, # of fires OFF, # of cycles)

This method will be implemented by counting the number of fires.

Two registers will define the “on” time where actual fire will occur (32bit), and two registers will define the “off” time where no jetting will happen.

For “on” period:

NUM\_OF\_FIRES\_ON\_L register, address TBD.

NUM\_OF\_FIRES\_ON\_H register, address TBD.

For “off” period:

NUM\_OF\_FIRES\_OFF\_L register, address TBD.

NUM\_OF\_FIRES\_OFF\_H register, address TBD.

The number of cycles will define in register NUM\_OF\_CYCL, address TBD

* + - 1. Single burst (# of fires) (256K fires)

Same as section 2.4.3.2 but with multiplier of 1.

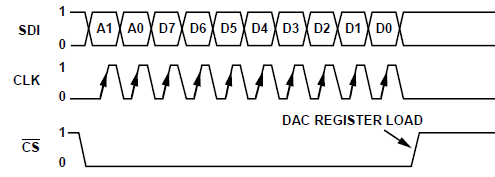
* + - 1. Firing for a set time duration

Switching bit TBD in WR\_DIAG2 register (address TBD) on and off after period of time wanted.

* 1. **Head voltage control**

The procedures of head voltage calibration, voltage & PS reading are the same as in E1 heads.

The calibration is done by SPI channel connected to digital potentiometer. At E1 head drive we are using p/n: AD8400AR1 – a single digital potentiometer. In GEN4 head drive, because we need to control on two voltages, we are using AD8402ARUZ1 - two digital potentiometer in single package. The data will need to be changed to fit the right potentiometer we want to control.



The /CS remains the same. To address Head1 and Head2 we use CS of Head1. Head2 /CS (as in E1) will be not in use (in case the head configuration will be 1, 3, 5, 7)

D9 (A1 in the picture above) and D8 (A0) control the address of the potentiometer.

D9, D8 = “00” – selecting potentiometer of head1.

D9, D8 = “01” - selecting potentiometer of head2.

* + 1. Calibration of head voltage (4 x 2 channels)

Same as in E1 procedure.

* + 1. Actual voltage reading

At E1 configuration, we are using only one analog reading of Vpp voltage. At GEN4 we need 2 analog readings, and this accomplish by using analog switcher/multiplexer.

Register TBD control Vpp reading:

D0=’0’ – selecting Vpp sense1.

D0=’1’ – selecting Vpp sense2.

* + 1. Error indication

Check and report by Main MCU.

* + 1. Heads PS voltage reading

Heads PS voltage reading is done in Main MCU on port AIN0. This is dedicated 8 analog inputs. This is the same as in E1 configuration.

* 1. **Resin-fill control**

All analog readings, including thermistors and analog sensors, is done by Main MCU and A2D MCU.

The analog path of thermistors and sensors are as follows:

|  |  |  |
| --- | --- | --- |
| Input name | HW signal name | A2D MCU port |
| M\_1\_FULL | AN13 | P1.4 |
| M\_1\_HFULL | AN14 | P1.5 |
| SUP\_FULL | AN15 | P1.6 |
| SUP\_HFULL | AN16 | P1.7 |
| M\_2\_FULL | AN17 | P2.4 |
| M\_2\_HFULL | AN18 | P2.5 |
| M\_3\_FULL | AN19 | P2.6 |
| M\_3\_HFULL | AN20 | P2.7 |
| TEMP1 | AN9 | P1.0 |
| TEMP2 | AN10 | P1.1 |
| TEMP3 | AN11 | P1.2 |
| TEMP4 | AN12 | P1.3 |
| TEMP5 | AN22 | P0.5 |
| VAC\_SNS | AN21 | P0.4 |

* + 1. Setting threshold values for all 4 sensors or for all 6 sensors in case where a “flooding“ reservoir is being used.

Controlled by Main MCU.

* + 1. Setting hysteresis value

Controlled by Main MCU.

* + 1. Enable/disable control
    2. Setting timeout period
    3. Timeout error indicator
    4. Pump Active indicator for every one of the materials

Pumps are controlled by MSC through OCB board. The application will need to control the pumps by sending proper commands to OCB board

* + 1. Setting of operating mode. Select between “flood“ mode and 4-material mode. In case of a“flood“ mode – selection of material source (pump assignment). (how the selecting done?)
  1. **Vacuum and purge control**

All vacuum, air and purge control is being done by OCB board.

* + 1. Vacuum valve On/Off control
    2. Setting purge duration
    3. Purge On/Off control
    4. Air Valve actuator
    5. Actual vacuum sensor readout

See section 2.6.

* 1. **Actuators** 
     1. Material pumps

Material pumps are controlled by MSC board through OCB board.

* + 1. LED illumination

Not supported

* 1. **Communication setup**