
1. Table of Contents

- 1. Table of Contents**
- 2. ASIC Overview**
- 3. Pin Diagram / Description**
- 4. Application Circuit / Functional Block Diagram**
- 5. Electrical Spec**
- 6. SPI Interface**
- 7. Normal Operating Modes and Description**
- 8. Test Modes**
- 9. IC Package Mechanical Drawing**
- 10. System Operational Flow Charts**
- 11. Revision History**
- 12. Appendix A**
- 13. Appendix B**
- 14. Appendix C**

2. ASIC Overview

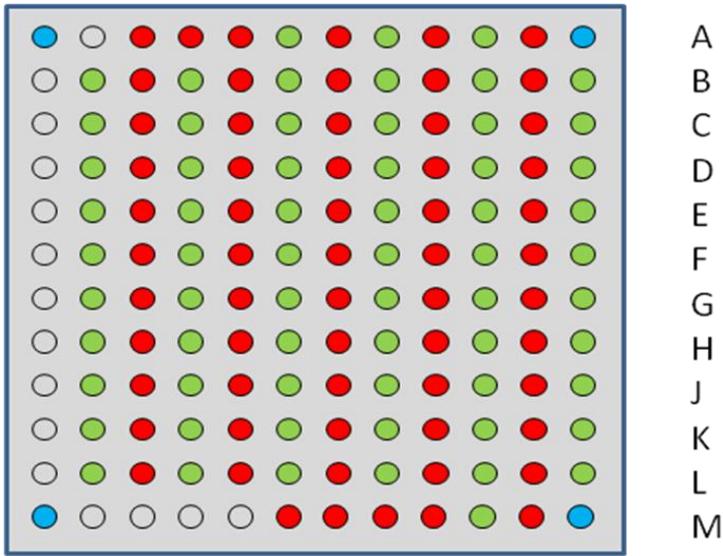
BFL SHA2 ASIC

- SHA-2 hardware (integrated circuit) Hash engine
- 65nm Technology
- BGA 10mm x 10mm, 144 ball package - solder bumps arranged in a 12 x 12 array, with a 0.8mm ball pitch
- Each Hash engine is comprised of 64 Unrolled or Pipelined stages x 2 (BFL architecture)
- 16 engines per chip
- Each hash engine is capable of running at typically 274M-Hash / sec, with a total chip combined hash rate of 4.2 G-hash / sec.
- At the rate of 4.2 G-Hash / sec, the ASIC dissipates approx 12.0 Watts
- All Hash engine operations / instructions / data are transmitted via a single SPI bus (4 pins) plus additional Done_In and Done_Out daisy chained interrupt pins
- SPI addresses 128 HASH machines (-- 3 bit chip address, 4 bit engine address, plus 8 bit register address and R/W bit for each Hash machine)
- Uses only 5 MPU I/O pins
- No glue logic required
- No device polling required – each “done” hash engine provides an unique vector address
- Double buffered registers can pre-load registers with new values while hash engine is processing previous register values



3. Pin Diagram / Pin Description

M10	VDDHV	I/O pwr	I/O vdd power
A11	VDDHV	I/O pwr	I/O vdd power
A1,A12 M1,M12	NC		unconnected
C12	SPI_CLK	input	SPI clock
D12	SPI_NCS	input	SPI Select, Active Low
E12	SPI_MOSI	input	SPI input
F12	SPI_MISO	TRI-out	SPI Tri-stated Output
G12	Done_In	input	Daisy Chain input for Hash complete bit *the farthest chip away from MCU connects to GND
H12	Done_Out	output	Status bit for Hash Complete, Daisy Chain connects to next chip's done in or MCU data pin
J12	Chip_Adrs_A0	input	3 bit unique chip address, address hard wired on the board
K12	Chip_Adrs_A1	input	3 bit unique chip address, address hard wired on the board
L12	Chip_Adrs_A2	input	3 bit unique chip address, address hard wired on the board
M11	ext_clk	input	External clock option -- for testing only
	VDD	pwr	All locations with Red marked balls (dark)
	VSS	pwr	All locations with Green marked balls (light)
B12	Test	output	Clock test out – Internal clock/64 – (Non divided version)
M9	Test	output	Temp P terminal
M8	Test	output	Temp N terminal



12 11 10 9 8 7 6 5 4 3 2 1

Fig. 1

4. Application Circuit / Functional Block Diagram

Application Circuit

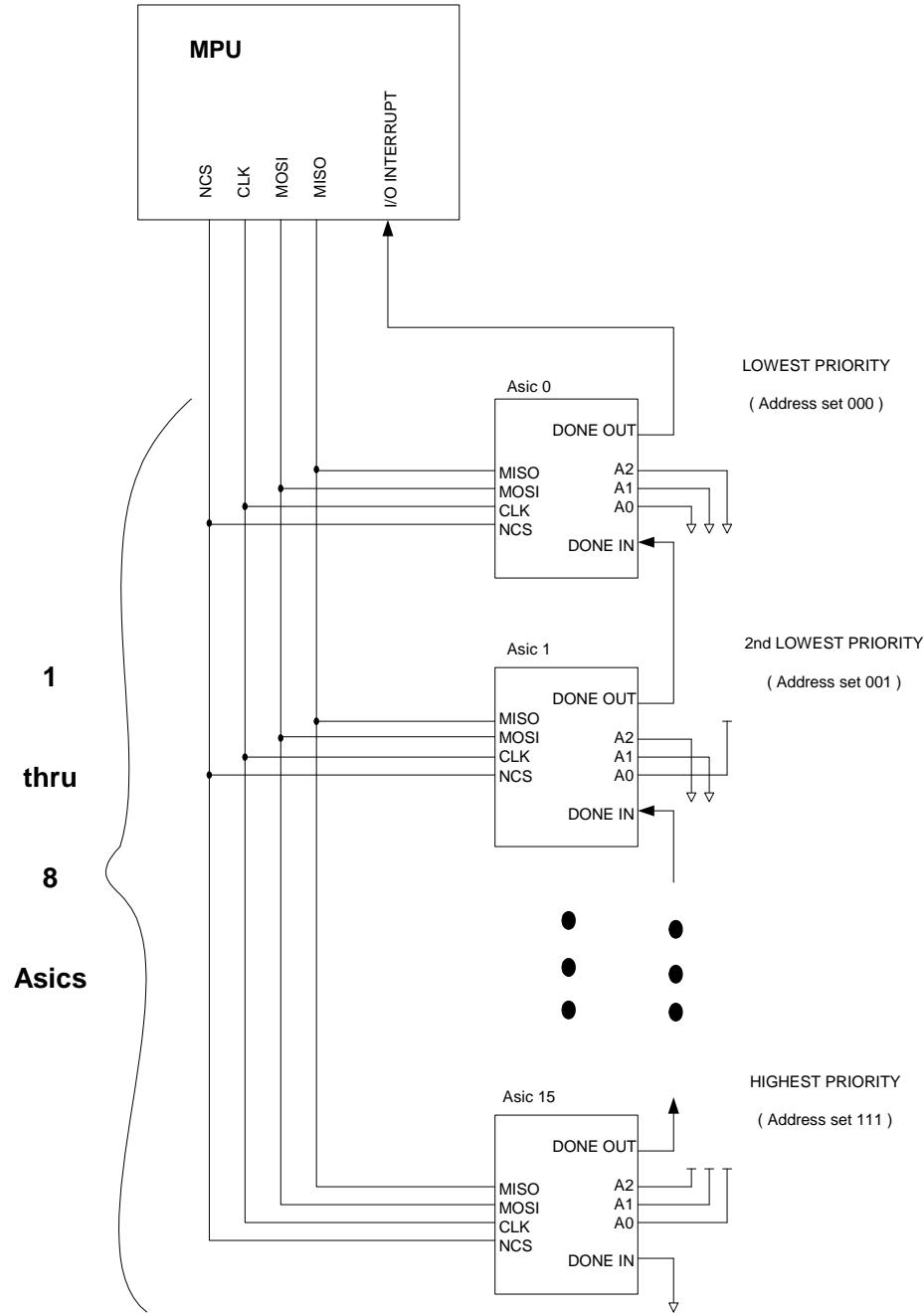


Fig.2



4.1 Application Circuit Description

The BFL SHA2 Asic can be combined with additional Asics up to a maximum of 8 Asics per system as shown in Fig2.

The chip addresses A(2:0) should be hardwired on the circuit board for the logic selection of each ASIC, and only one ASIC should be programmed with any specific address.

Further selection of SHA2 engines takes place within each ASIC, where a selection of one 1 from 16 engines can be selected by writing or reading from a specific SPI address PA(3:0). This is further described in fig3. the ASIC functional block diagram.

Within an ASIC the control to enable the clock to each ASIC plus additional test functions for clock and frequency control for all instances within an ASIC are handled by engine '0', the engines being designated 0 – 15 in each ASIC. Specific operational commands to reset each engine, initiate a computation, or read back the results of a specific instance computation are handled by the SPI interface specific to that particular engine.

Since multiple engines within an ASIC will likely be operating simultaneously, a method to recognize when a computation has been initiated or has been completed is necessary. When an engine is busy it sets an SPI register with this status, and when the computation is completed a flag indicating the completion is also set. Engine completions are or'd into a single status back to the uP from the ASIC and it is expected that the uP will poll this status line to learn when a computation has been completed. If multiple completions occur simultaneously, a priority system has been include in the interface between engine instances on an ASIC to ensure that no loss of result data will occur as a result.

A block diagram of the functional interaction between the various instances described above is shown in the Functional Block Diagram Fig. 3 on the next page.



Functional Block Diagram

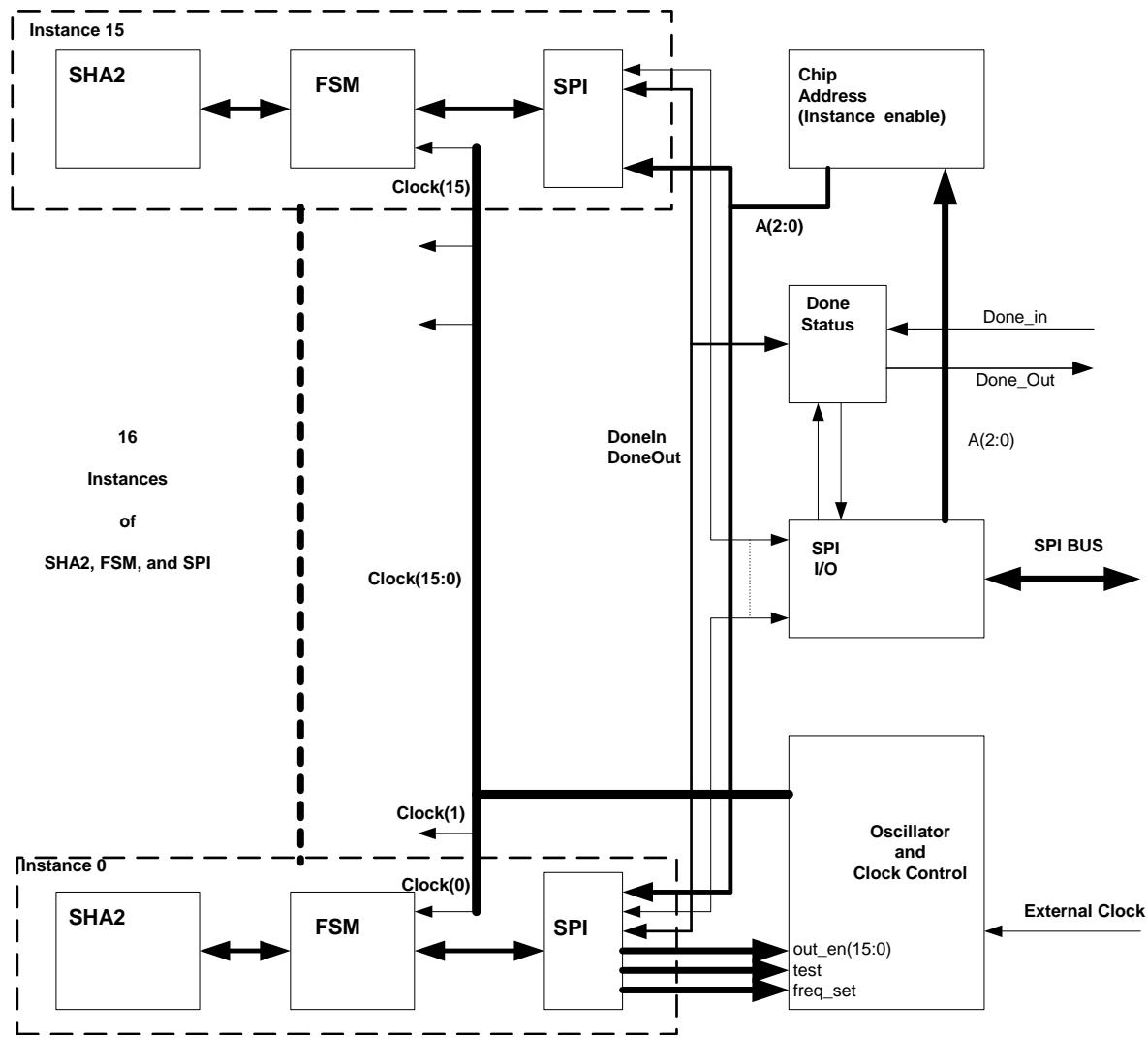


Fig.3

The external clock input should be pulled to 0V on the board via a pull down resistor. This input is for test purposes only and will not be used during normal operation.



FSM Controller

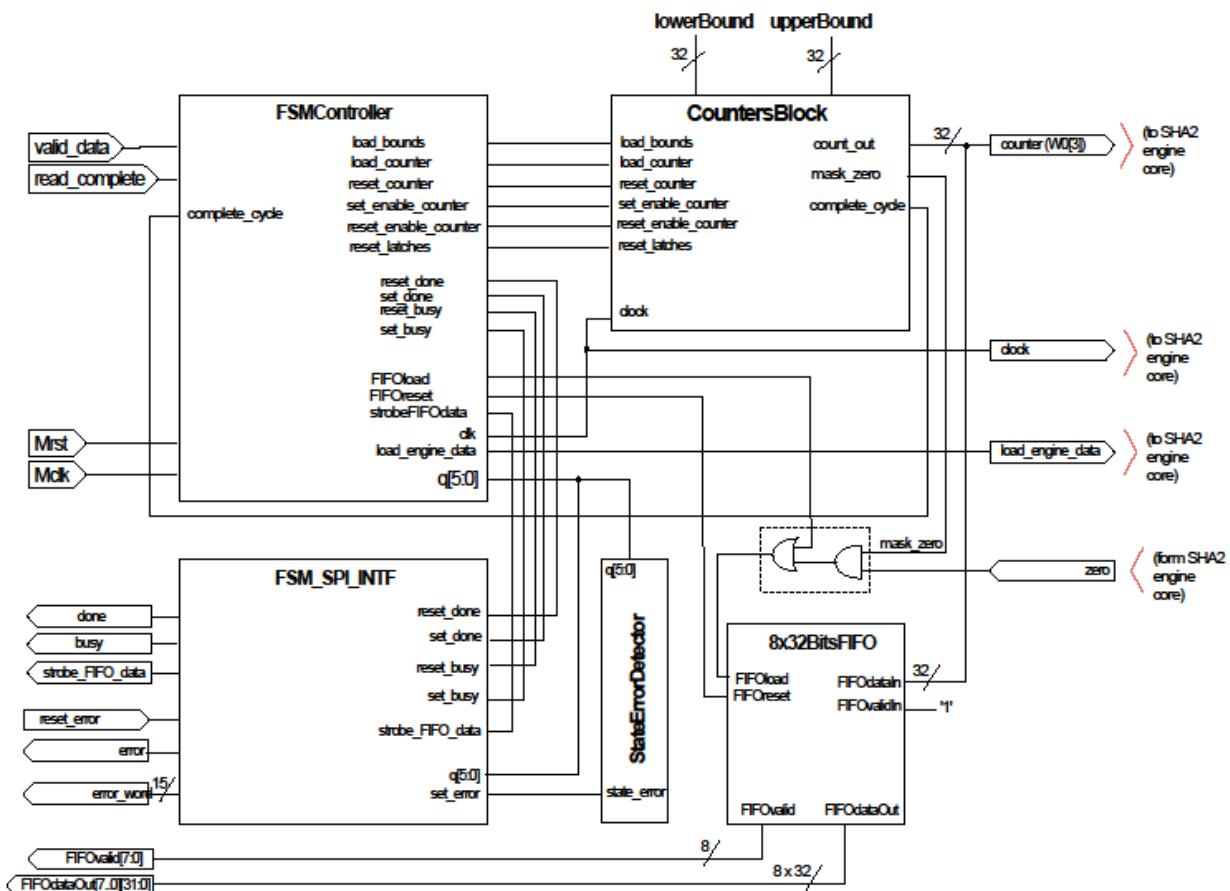


Fig.4

The FSM controller is the primary interface between the SHA2 engine and the SPI interface out to the uProcessor. This block regulates requests from the up via the SPI interface and communicates completion status, busy status, and any operational status required as to the various state machines controlling the engine computation.

Additionally during computation activity, intermediate computation results from the engine are stored temporarily in a local FIFO prior to transferring to the SPI interface for readout by the uProcessor.



5. Electrical Spec

Chip Power Supply

** Use Ceramic Decoupling Caps of 4.7uF or greater, between power and ground pin, using no VIA on PCB.

Absolute maximum ratings

PARAMETER	SYMBOL	RATINGS	UNITS
Supply Voltage I/O	Vdd I/O	3.6	Volt
Supply Voltage Core	Vdd core	1.21	Volt
ESD tolerance, HBM	Vesd	2000	Volt
Operating Ambient temp. Range	To	15 to 55	Celcius
Storage temp. range	Ts	-40 to +85	Celsius

Operating conditions

PARAMETER	SYMBOL	TEST CONDITION	MIN	TYP	MAX	UNIT
Supply Voltage I/O	Vdd I/O		3	3.3	3.6	V
Supply Voltage core	Vdd core		0.95	1	1.05	V
Supply Current I/O	I I/O				100	mA
Supply Current core	I core			10		A
Power Dissipation	Pd			10		Watts
Standby current	I standby	All clocks disabled		tbd		mA
SPI_CLK frequency					10	MHz



6. SPI Interface

Read cycle timing diagram

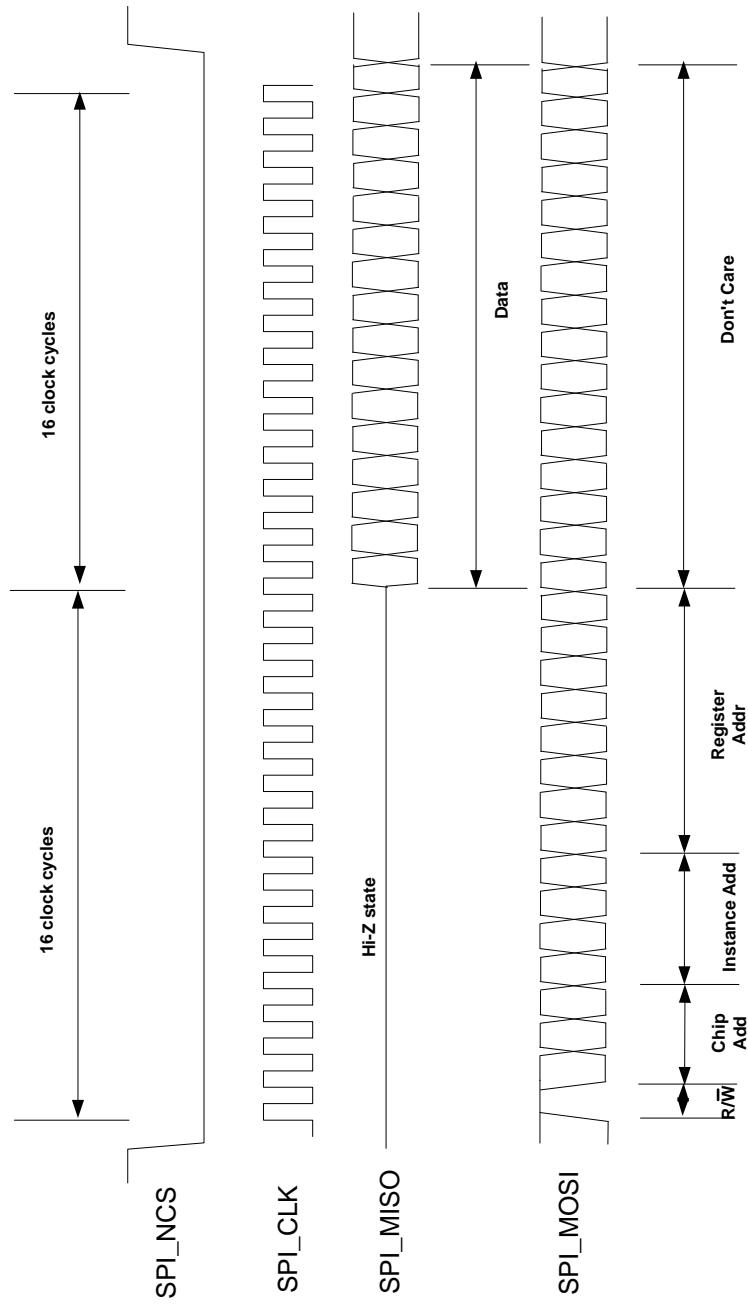


Fig.5



Write cycle timing diagram

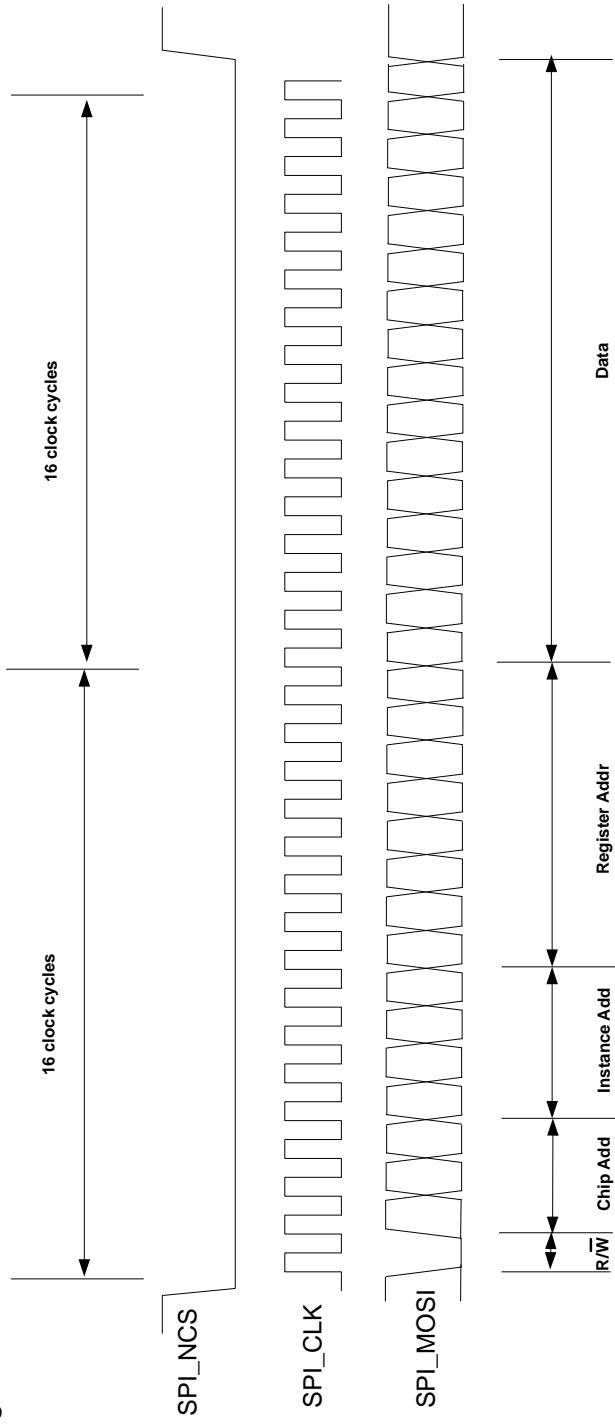


Fig.6



7. Normal Operating Modes and Description

Pins used for normal operations –

3 chip address pins, hard soldered to PCB (for max of 8 ASICs per PCB).

2 pins for Daisy chain interrupt, 1 input (Done_In), 1 output (Done_Out) (see application diagram for systems connections)

1 pin for external clock, normally grounded

4 SPI bus pins (Mode 00 operation, 32 bit word; can be done with 2 x 16bit transfers)

SPI

For SPI synchronization after power up, SPI_NCS must be set high with at least one positive edge transition of SPI_CLK

SPI_NCS must not be tied low for proper chip function (because doneout/donein are clocked on the negative edge of SPI_NCS).

Once SPI has been synchronized, there are a number of steps that need to be taken to initialize the chip for normal operation.

The internal chip oscillator should be set to its normal operating mode via the SPI interface.

	Address	Data
Enable oscillator	Oddd0000 00000000	0000000x xxxxxxx1
Set Oscillator frequency	Oddd0000 01100000	11111111 11010101

The 16 data bits of register address 01100001 each enable the clock for one of the 16 chip hash engines. Set the data bit to "1" for all Hash Engines for which the clock should be enabled

Set clk out enables Oddd0000 01100001 dddddddd dddddddd



At this point the internal oscillator should be running and clocks should be enabled for all required engines. For normal operation of each engine follow the instructions in the section 10 "System Operational Flow Charts" later in this document.

Note that on the ASIC, control for the oscillator and clocking is common for all instances and is done one time using the Instance0 SPI interface. All other controls for each instance (including Reset) are controlled individually using the SPI interface for that specific instance.

ADDRESSING

There are 16 Hash engines per chip, up to 8 chips per PCB, please refer to the SPI register map for the 7 bit address assignments for each of the total 128 engines.

DONE INTERRUPT

The done signal is daisy chained through all 128 machines (8 chips x 16 hash engines per chip), the engine farthest from the MCU (as wired on the PCB) has the highest priority.

When an Engine is done, it will signal the MCU via the Done_Out pin, the MCU then should send out the Global Query signal, after which only the Hash engine with a set "DoneCycle" with highest priority will load its 7 bit address on the SPI bus for the MCU to read. Thus the MCU can subsequently address that engine to read its results (in the fifo registers). The read status register (see Appendix A) indicates if there are any new counter values stored in the fifo.

PIPELINING

Once an Engine is running (Busy - see appendix A), it can be preloaded with the next values which will run immediately as the previous run completes.



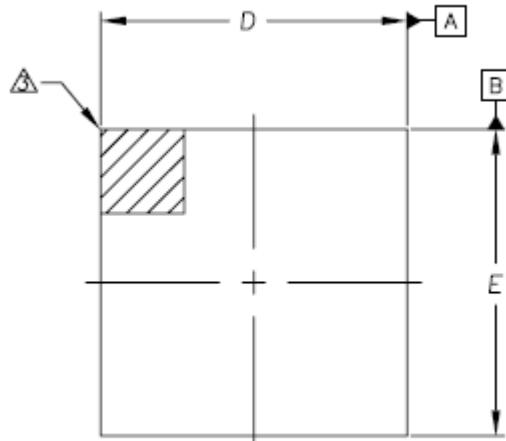
8. Test Modes

Blank

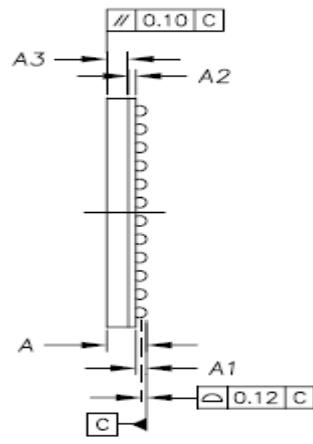


9. IC Package Mechanical Drawing

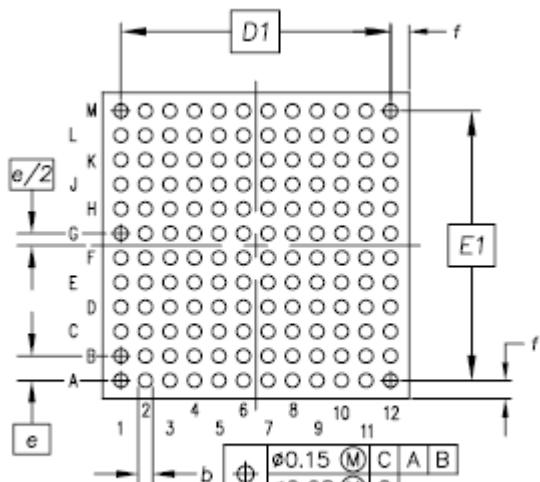
BGA – 10mm x 10mm
Pin Count -- 144 pin



TOP VIEW



SIDE VIEW



BOTTOM VIEW

SYMBOL	MILLIMETER			INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.20	1.30	1.40	.047	.051	.055
A1	0.30	0.35	0.40	.012	.014	.016
A2	0.20	0.25	0.30	.008	.010	.012
A3	0.65	0.70	0.75	.026	.028	.030
b	0.40	0.45	0.50	.016	.018	.020
D	9.90	10.00	10.10	.390	.394	.398
D1	8.80 BSC			.346 BSC		
E	9.90	10.00	10.10	.390	.394	.398
E1	8.80 BSC			.346 BSC		
e	0.80 BSC			.031 BSC		
f	0.50	0.60	0.70	.020	.024	.028



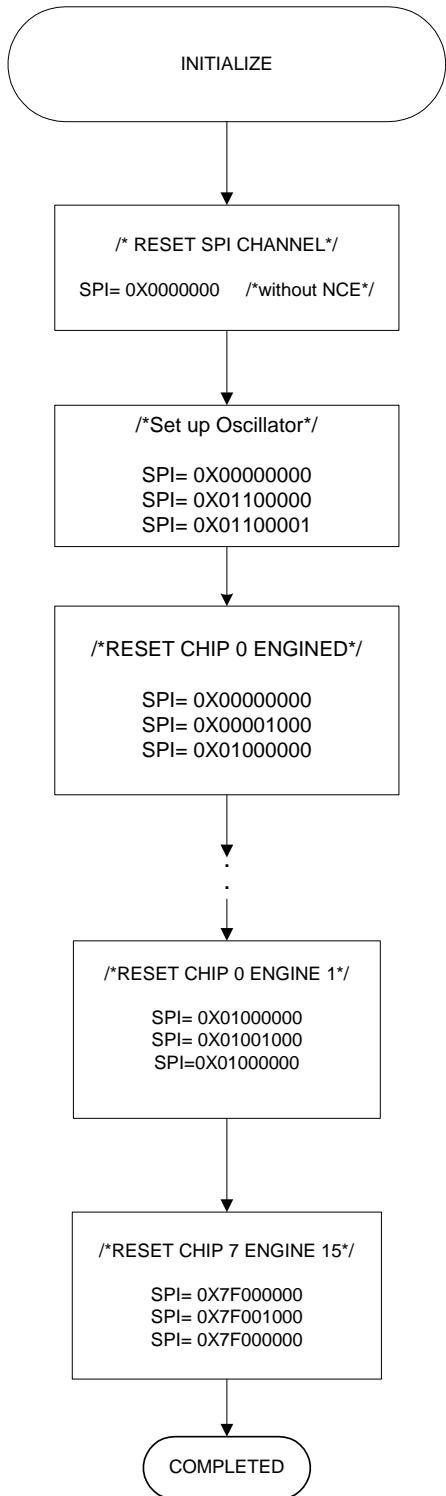
10. SYSTEM OPERATIONAL FLOW CHARTS

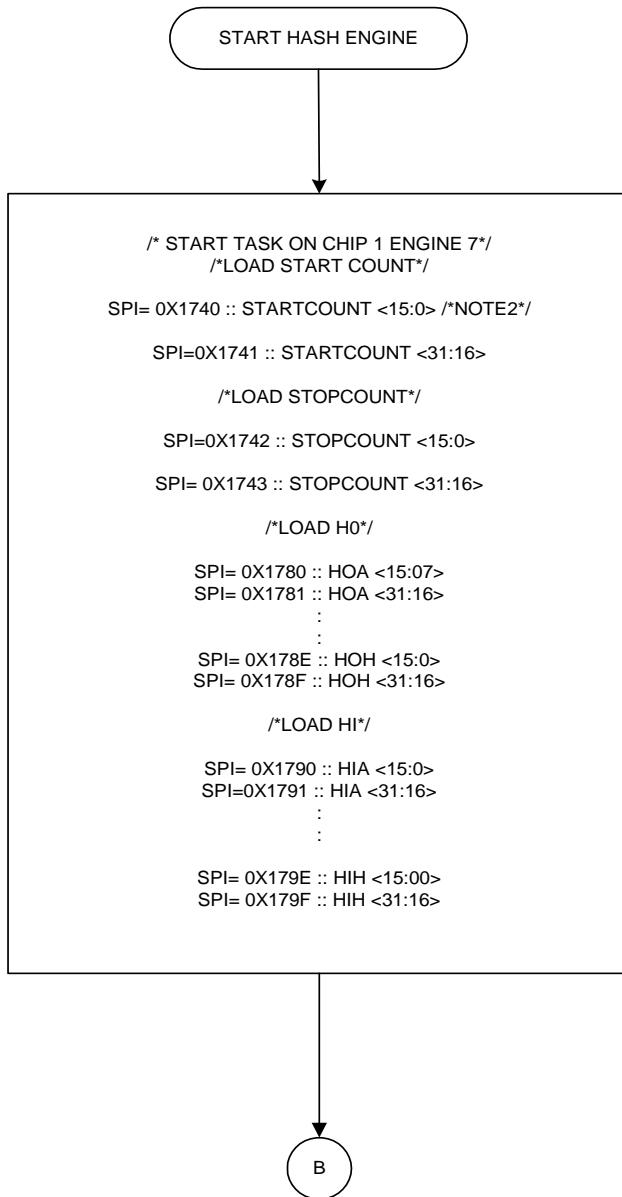
The operation of each engine can be described by a number of flow charts outlining the steps that should be taken to prepare the engine for operation after power on. This detail is contained in the "Initialize" flow chart.

After successful power up and engine initialization, a second flow chart "Start Hash Engine" flow chart describes the necessary steps to initiate a hash engine computation.

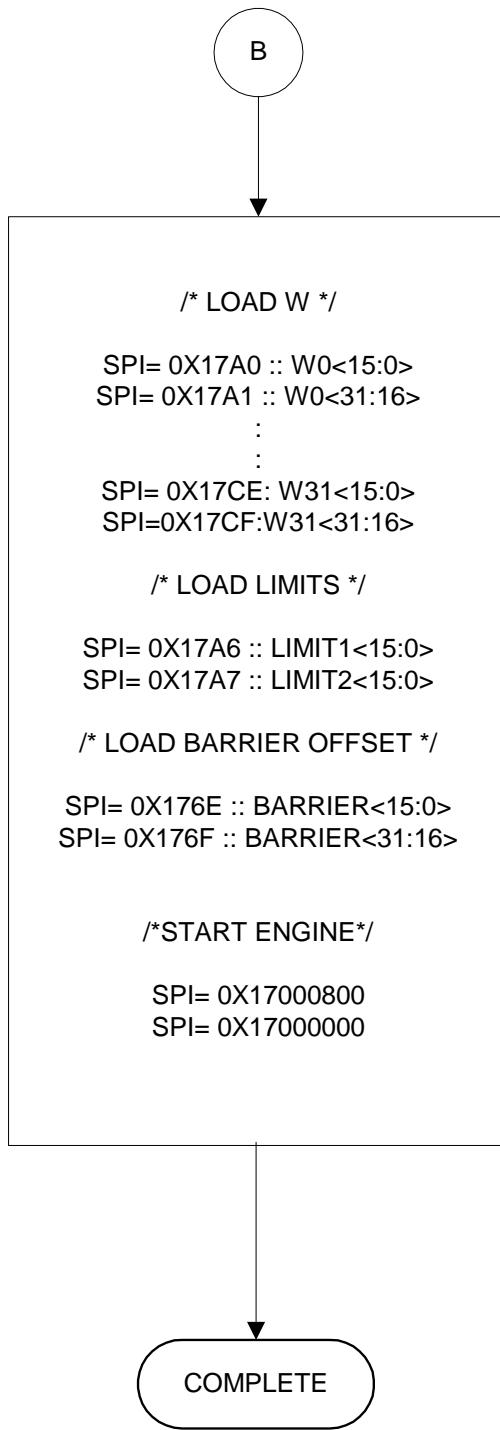
Lastly after the computation has been started a third flow chart "Awaiting Processing Completion" flow chart describes the steps that are required while waiting for a computation to be completed and how to recognize when the Hash Engine has completed the computation and is available for additional computations

Initialize Flow Chart

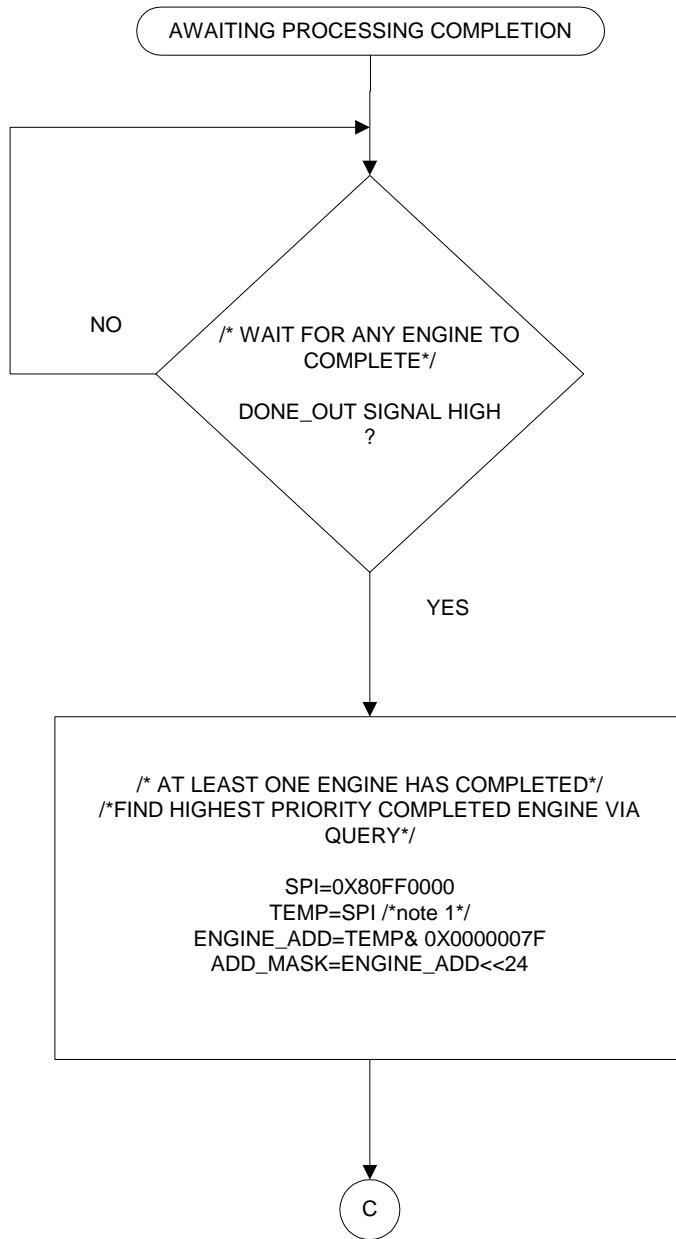


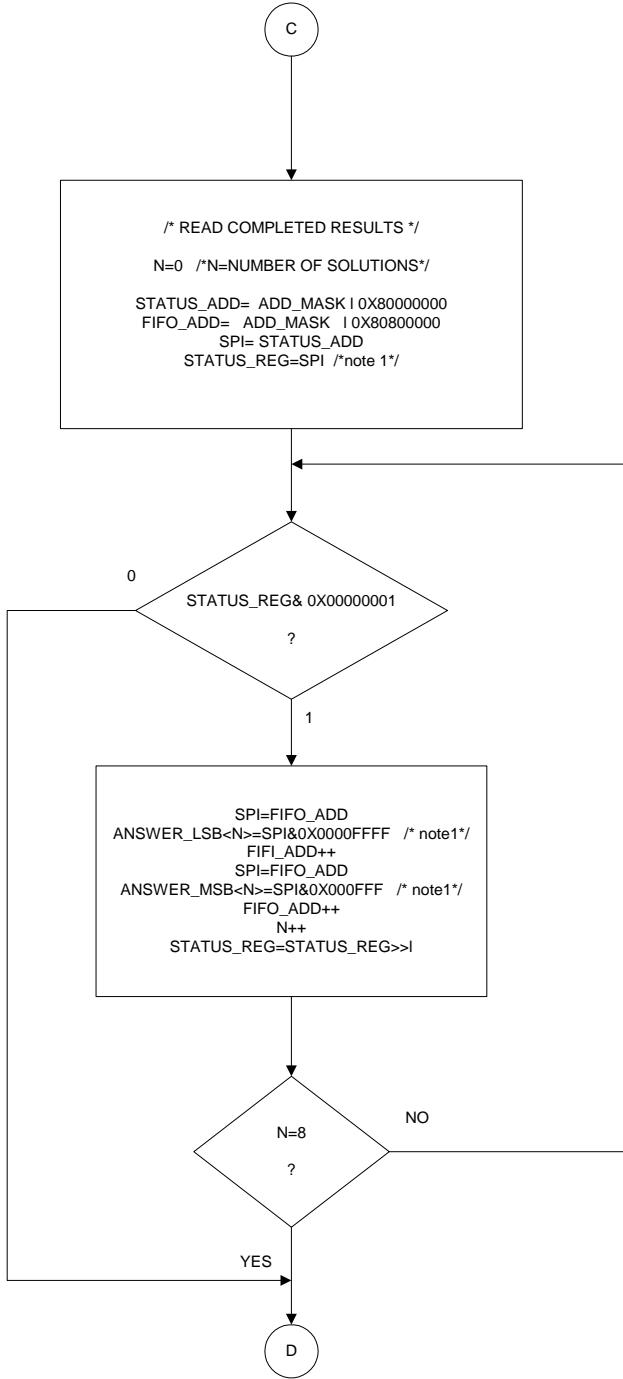
Start Hash Engine
Flow Chart

NOTE 2 " :: " DENOTES CONCATENATE TWO 16 BIT VALUES INTO ONE 32 BIT VALUE
START COUNT <15:0> DENOTES LOW 16 BITS OF START COUNT
STARTCOUNT <31:16> DENOTES UPPER 16 BITS OF STARTCOUNT

Start Hash Engine
Flow Chart – cont.

Hash Engine Processing awaiting completion

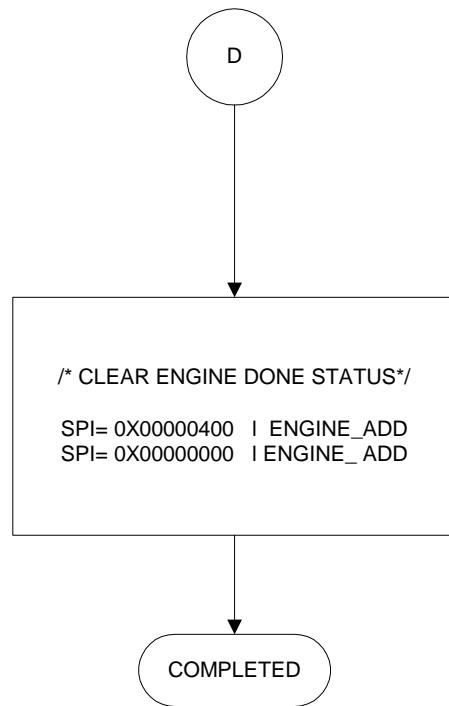


Hash Engine Processing
 awaiting completion – cont.


NOTE 1: SPI READS MUST WAIT UNTIL SPI WRITE CYCLE IS COMPLETED



Hash Engine Processing awaiting completion – cont.





BUTTERFLYLABS

65nm Chip Specifications
BFL SHA2 ASIC

11. Revision History

Page | 22

Product specifications are subject to change without notice. No responsibility is assumed for the use of information herein

12. Appendix A

The individual Hash engine register map and register descriptions is shown on the following pages .

Each Hash engine has a complete set of individual registers which completely control the operation of that particular machine with one exception.

The registers which control the operation of the oscillator for the chip can only be controlled by writing to Hash engine instance 0 of the chip

For all Hash engines Write Control Register bits d10, d11, and d12 control the documented functions as outlined in the register map. Only Hash engine instance 0 controls the the additional oscillator control functions documented in the register map

Additionally only the Oscillator Control register, and the Clock Out Enable register of instance 0 control the Oscillator frequency and clock enable functions for all of the Hash engine instances.



65nm Chip Specifications

BFL SHA2 ASIC

BFL Register Map SPI MODE 00						
	32 bit SPI WORD					
Description	Signal Name	R/W	chip	engine	spi - 8 bit	d=data x=don't care
Denotes a Global Register which controls all 16 Cores						
Denotes a Local Register which controls only 1 Core						

Write Register Section

1	Write Control Register							
	Oscillator control **** 000 only, for whole chip							
	external clock	(MSB) d15	osc_sel_ext	0	ddd	0000 0000	Dddd dddx xxdd dddd	
	divide by 2 --- 2x slower ie. 250 Mhz	d14	osc_div2	0	ddd	0000 0000	dDdd dddx xxdd dddd	
	divide by 4 --- 4x slower ie. 125Mhz	d13	osc_div4	0	ddd	0000 0000	ddDd dddx xxdd dddd	
	Reset	d12	Reset	0	ddd	dddd 0000 0000	dddX dddx xxdd dddd	
	WriteRegistersValid	d11	WriteRegistersValid	0	ddd	dddd 0000 0000	dddd Dddx xxdd dddd	
	ReadCompleted	d10	ReadCompleted	0	ddd	dddd 0000 0000	dddd dDdx xxdd dddd	
	Reset_Error_SPI	d9	ResetError_spi	0	ddd	dddd 0000 0000	dddd ddDx xxdd dddd	
		d8	not used	0	ddd	dddd 0000 0000	dddd dddX xxdd dddd	
		d7	not used	0	ddd	dddd 0000 0000	dddd dddx Xxdd dddd	
		d6	not used	0	ddd	dddd 0000 0000	dddd dddx xXdd dddd	
	not used	d5	not used	0	ddd	0000 0000 0000 0000	dddd dddx xxDd dddd	
	not used	d4	not used	0	ddd	0000 0000 0000 0000	dddd dddx xxdD dddd	
	not used	d3	not used	0	ddd	0000 0000 0000 0000	dddd dddx xxdd Dddd	
	not used	d2	not used	0	ddd	0000 0000 0000 0000	dddd dddx xxdd dDdd	
	not used	d1	not used	0	ddd	0000 0000 0000 0000	dddd dddx xxdd ddDd	
	osc_div1	(LSB) d0	osc_div1	0	ddd	0000 0000 0000 0000	dddd dddx xxdd dddD	

2	Test SPI BUS write		0	ddd	dddd	0000 0001	dddd dddd dddd dddd	
3	lower count -- lsb		0	ddd	dddd	0100 0000	Count Start <31:0> lower 16bits	
4	lower count -- msb		0	ddd	dddd	0100 0001	Count Start <31:0> upper 16bits	
5	upper count -- lsb		0	ddd	dddd	0100 0010	Count End <31:0> lower 16bits	
6	upper count -- msb		0	ddd	dddd	0100 0011	Count End <31:0> upper 16bits	
7	Oscillator Control	Oscillator Control	0	ddd	dddd	0110 0000	All 16 bits	
8	Reserve for chip Test via SPI bus	Clk_out_enable	0	ddd	dddd	0110 0001	All 16 bits	
9	Reserve for chip Test via SPI bus		0	ddd	dddd	0110 0010		
10	Reserve for chip Test via SPI bus		0	ddd	dddd	0110 0011		
11	Reserve for chip Test via SPI bus		0	ddd	dddd	0110 0100		
12	Reserve for chip Test via SPI bus		0	ddd	dddd	0110 0101		
13	Reserve for chip Test via SPI bus		0	ddd	dddd	0110 0110		
14	Reserve for chip Test via SPI bus		0	ddd	dddd	0110 0111		
15	Reserve for chip Test via SPI bus		0	ddd	dddd	0110 1000		
16	Reserve for chip Test via SPI bus		0	ddd	dddd	0110 1001		
17	Reserve for chip Test via SPI bus		0	ddd	dddd	0110 1010		
18	Reserve for chip Test via SPI bus		0	ddd	dddd	0110 1011		
19	Reserve for chip Test via SPI bus		0	ddd	dddd	0110 1100		
20	Reserve for chip Test via SPI bus		0	ddd	dddd	0110 1101		
21	Barrier Offset		0	ddd	dddd	0110 1110	Barrier Offset<31:0> low 16bits	
22	Barrier Offset		0	ddd	dddd	0110 1111	Barrier Offset<31:0> up 16bits	
15	First sha2_64 H_A lower 16 bits input		0	ddd	dddd	1000 0000	H0 A<31:0> lower 16bits	
16	First sha2_64 H_A upper 16 bits input		0	ddd	dddd	1000 0001	H0 A<31:0> upper 16bits	
17			0	ddd	dddd	1000 0010	H0 B<31:0> lower 16bits	
18			0	ddd	dddd	1000 0011	H0 B<31:0> upper 16bits	
19			0	ddd	dddd	1000 0100	H0 C<31:0> lower 16bits	
20			0	ddd	dddd	1000 0101	H0 C<31:0> upper 16bits	
21			0	ddd	dddd	1000 0110	H0 D<31:0> lower 16bits	
22			0	ddd	dddd	1000 0111	H0 D<31:0> upper 16bits	
23			0	ddd	dddd	1000 1000	H0 E<31:0> lower 16bits	
24			0	ddd	dddd	1000 1001	H0 E<31:0> upper 16bits	
25			0	ddd	dddd	1000 1010	H0 F<31:0> lower 16bits	
26			0	ddd	dddd	1000 1011	H0 F<31:0> upper 16bits	
27			0	ddd	dddd	1000 1100	H0 G<31:0> lower 16bits	

28		0	ddd	dddd	1000 1101	H0 G<31:0> upper 16bits
29		0	ddd	dddd	1000 1110	H0 H<31:0> lower 16bits
30		0	ddd	dddd	1000 1111	H0 H<31:0> upper 16bits
31	Second sha2_64 H_A lower 16 bits input	0	ddd	dddd	1001 0000	H1 A<31:0> lower 16bits
32	Second sha2_64 H_A upper 16 bits input	0	ddd	dddd	1001 0001	H1 A<31:0> upper 16bits
33		0	ddd	dddd	1001 0010	H1 B<31:0> lower 16bits
34		0	ddd	dddd	1001 0011	H1 B<31:0> upper 16bits
35		0	ddd	dddd	1001 0100	H1 C<31:0> lower 16bits
36		0	ddd	dddd	1001 0101	H1 C<31:0> upper 16bits
37		0	ddd	dddd	1001 0110	H1 D<31:0> lower 16bits
38		0	ddd	dddd	1001 0111	H1 D<31:0> upper 16bits
39		0	ddd	dddd	1001 1000	H1 E<31:0> lower 16bits
40		0	ddd	dddd	1001 1001	H1 E<31:0> upper 16bits
41		0	ddd	dddd	1001 1010	H1 F<31:0> lower 16bits
42		0	ddd	dddd	1001 1011	H1 F<31:0> upper 16bits
43		0	ddd	dddd	1001 1100	H1 G<31:0> lower 16bits
44		0	ddd	dddd	1001 1101	H1 G<31:0> upper 16bits
45		0	ddd	dddd	1001 1110	H1 H<31:0> lower 16bits
46		0	ddd	dddd	1001 1111	H1 H<31:0> upper 16bits
47		0	ddd	dddd	1010 0000	W0<31:0> lower 16bits
48		0	ddd	dddd	1010 0001	W0<31:0> upper 16bits
49		0	ddd	dddd	1010 0010	W1<31:0> lower 16bits
50		0	ddd	dddd	1010 0011	W1<31:0> upper 16bits
51		0	ddd	dddd	1010 0100	W2<31:0> lower 16bits
52		0	ddd	dddd	1010 0101	W2<31:0> upper 16bits
53	Limit 1	0	ddd	dddd	1010 0110	Limit<31:0> lower 16bits
54	Limit 2	0	ddd	dddd	1010 0111	Limit<31:0> lower 16bits
55		0	ddd	dddd	1010 1000	W4<31:0> lower 16bits
56		0	ddd	dddd	1010 1001	W4<31:0> upper 16bits
57		0	ddd	dddd	1010 1010	W5<31:0> lower 16bits
58		0	ddd	dddd	1010 1011	W5<31:0> upper 16bits
59		0	ddd	dddd	1010 1100	W6<31:0> lower 16bits
60		0	ddd	dddd	1010 1101	W6<31:0> upper 16bits
61		0	ddd	dddd	1010 1110	W7<31:0> lower 16bits
62		0	ddd	dddd	1010 1111	W7<31:0> upper 16bits
63		0	ddd	dddd	1011 0000	W8<31:0> lower 16bits
64		0	ddd	dddd	1011 0001	W8<31:0> upper 16bits
65		0	ddd	dddd	1011 0010	W9<31:0> lower 16bits

66					0	ddd	dddd	1011 0011	W9<31:0> upper 16bits
67					0	ddd	dddd	1011 0100	W10<31:0> lower 16bits
68					0	ddd	dddd	1011 0101	W10<31:0> upper 16bits
69					0	ddd	dddd	1011 0110	W11<31:0> lower 16bits
70					0	ddd	dddd	1011 0111	W11<31:0> upper 16bits
71					0	ddd	dddd	1011 1000	W12<31:0> lower 16bits
72					0	ddd	dddd	1011 1001	W12<31:0> upper 16bits
73					0	ddd	dddd	1011 1010	W13<31:0> lower 16bits
74					0	ddd	dddd	1011 1011	W13<31:0> upper 16bits
75					0	ddd	dddd	1011 1100	W14<31:0> lower 16bits
76					0	ddd	dddd	1011 1101	W14<31:0> upper 16bits
77					0	ddd	dddd	1011 1110	W15<31:0> lower 16bits
78					0	ddd	dddd	1011 1111	W15<31:0> upper 16bits
79					0	ddd	dddd	1100 0000	W24<31:0> lower 16bits
80					0	ddd	dddd	1100 0001	W24<31:0> upper 16bits
81					0	ddd	dddd	1100 0010	W25<31:0> lower 16bits
82					0	ddd	dddd	1100 0011	W25<31:0> upper 16bits
83					0	ddd	dddd	1100 0100	W26<31:0> lower 16bits
84					0	ddd	dddd	1100 0101	W26<31:0> upper 16bits
85					0	ddd	dddd	1100 0110	W27<31:0> lower 16bits
86					0	ddd	dddd	1100 0111	W27<31:0> upper 16bits
87					0	ddd	dddd	1100 1000	W28<31:0> lower 16bits
88					0	ddd	dddd	1100 1001	W28<31:0> upper 16bits
89					0	ddd	dddd	1100 1010	W29<31:0> lower 16bits
90					0	ddd	dddd	1100 1011	W29<31:0> upper 16bits
91					0	ddd	dddd	1100 1100	W30<31:0> lower 16bits
92					0	ddd	dddd	1100 1101	W30<31:0> upper 16bits
93					0	ddd	dddd	1100 1110	W31<31:0> lower 16bits
94					0	ddd	dddd	1100 1111	W31<31:0> upper 16bits

Read Register Section

1	Read Status Register								
	FIFO Depth 1	d15			1	ddd	dddd	0000 0000	Dddd dddd xxxx xddd
	FIFO Depth 2	d14			1	ddd	dddd	0000 0000	dDdd dddd xxxx xddd
	FIFO Depth 3	d13			1	ddd	dddd	0000 0000	ddDd dddd xxxx xddd
	FIFO Depth 4	d12			1	ddd	dddd	0000 0000	dddD dddd xxxx xddd
	FIFO Depth 5	d11			1	ddd	dddd	0000 0000	dddd Dddd xxxx xddd

	FIFO Depth 6	d10	1	ddd	dddd	0000 0000	dddd dDdd xxxx xddd	
	FIFO Depth 7	d9	1	ddd	dddd	0000 0000	dddd ddDd xxxx xddd	
	FIFO Depth 8	d8	1	ddd	dddd	0000 0000	dddd dddD xxxx xddd	
	not used	d7	1	ddd	dddd	0000 0000	dddd dddd Xxxx xddd	
	not used	d6	1	ddd	dddd	0000 0000	dddd dddd xXXX xddd	
	not used	d5	1	ddd	dddd	0000 0000	dddd dddd xxXx xddd	
	not used	d4	1	ddd	dddd	0000 0000	dddd dddd xxxX xddd	
	not used	d3	1	ddd	dddd	0000 0000	dddd dddd xxxx Xddd	
	not used	d2	1	ddd	dddd	0000 0000	dddd dddd xxxx xXdd	
	Busy	d1	1	ddd	dddd	0000 0000	dddd dddd xxxx xdXd	
	DoneCycle	d0	1	ddd	dddd	0000 0000	dddd dddd xxxx xddD	
2	Test SPI BUS read		1	ddd	dddd	0000 0001	dddd dddd dddd dddd	
3	FSM Status Word 0	d15	error_flag	1	ddd	dddd	0110 0000	Dddd dddd xxxx xddd
		d14	reset_error_FSM	1	ddd	dddd	0110 0000	dDdd dddd xxxx xddd
		d13	reset_done	1	ddd	dddd	0110 0000	ddDd dddd xxxx xddd
		d12	set_done	1	ddd	dddd	0110 0000	dddD dddd xxxx xddd
		d11	reset_busy	1	ddd	dddd	0110 0000	dddd Dddd xxxx xddd
		d10	set_busy	1	ddd	dddd	0110 0000	dddd dDdd xxxx xddd
		d9	FIFO_load	1	ddd	dddd	0110 0000	dddd ddDd xxxx xddd
		d8	FIFO_reset	1	ddd	dddd	0110 0000	dddd dddD xxxx xddd
		d7	FIFO_strobe	1	ddd	dddd	0110 0000	dddd dddd Xxxx xddd
		d6	limit_out1	1	ddd	dddd	0110 0000	dddd dddd xXXX xddd
		d5	limit_out2	1	ddd	dddd	0110 0000	dddd dddd xxXx xddd
		d4	mask_clock	1	ddd	dddd	0110 0000	dddd dddd xxxX xddd
		d3	clock	1	ddd	dddd	0110 0000	dddd dddd xxxx Xddd
		d2	clock_bar	1	ddd	dddd	0110 0000	dddd dddd xxxx xXdd
		d1	master_load	1	ddd	dddd	0110 0000	dddd dddd xxxx xdXd
		d0	master_reset	1	ddd	dddd	0110 0000	dddd dddd xxxx xddD
4	FSM Status Word 1	d15	read_state[3]	1	ddd	dddd	0110 0001	Dddd dddd xxxx xddd
		d14	read_state[2]	1	ddd	dddd	0110 0001	dDdd dddd xxxx xddd
		d13	read_state[1]	1	ddd	dddd	0110 0001	ddDd dddd xxxx xddd
		d12	read_state[0]	1	ddd	dddd	0110 0001	dddD dddd xxxx xddd
		d11	valid_state[3]	1	ddd	dddd	0110 0001	dddd Dddd xxxx xddd
		d10	valid_state[2]	1	ddd	dddd	0110 0001	dddd dDdd xxxx xddd
		d9	valid_state[1]	1	ddd	dddd	0110 0001	dddd ddDd xxxx xddd
		d8	valid_state[0]	1	ddd	dddd	0110 0001	dddd dddD xxxx xddd

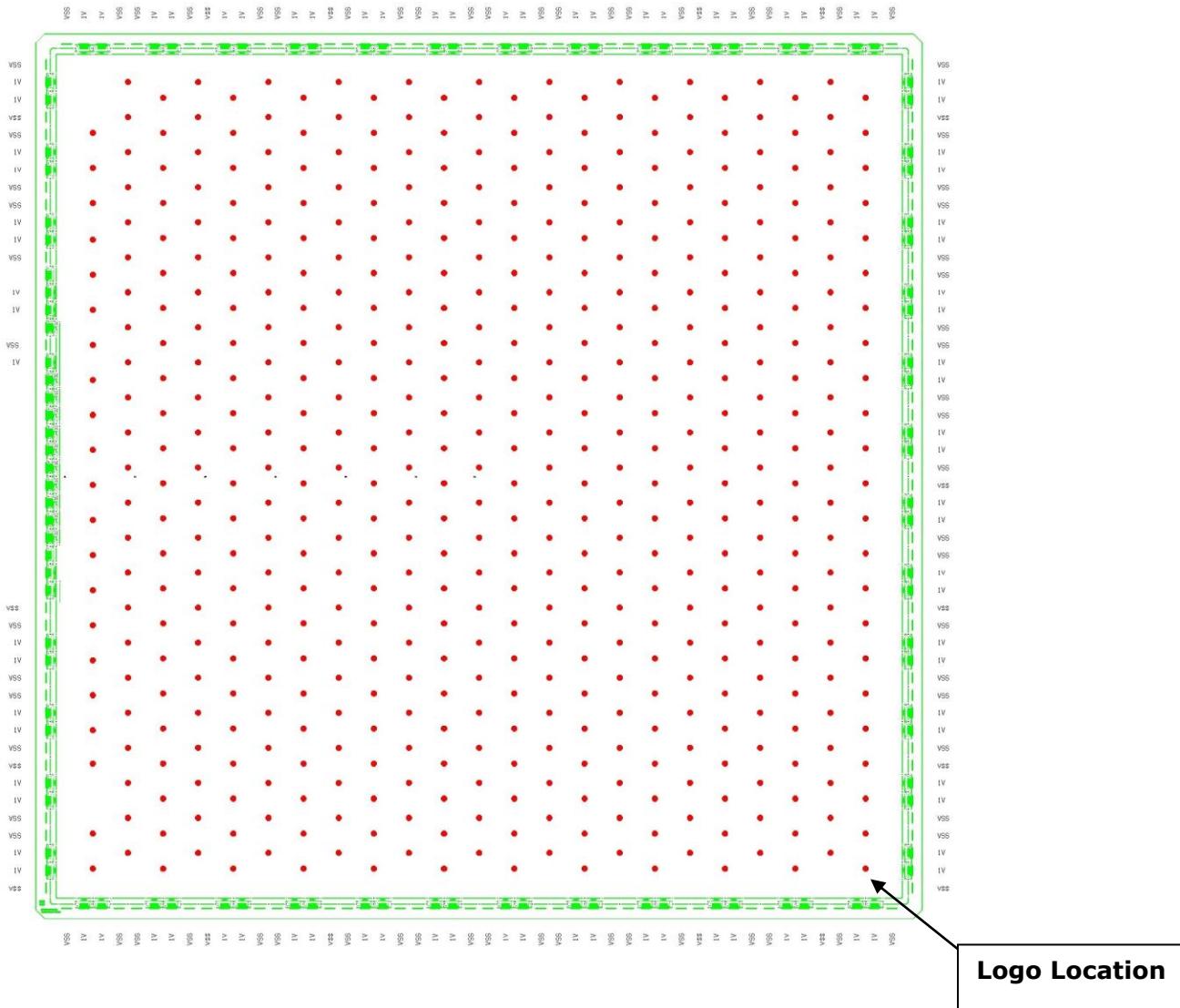
	d7	base_state[7]	1	ddd	dddd	0110 0001	dddd dddd Xxxx xddd
	d6	base_state[6]	1	ddd	dddd	0110 0001	dddd dddd xXXX xddd
	d5	base_state[5]	1	ddd	dddd	0110 0001	dddd dddd xxXx xddd
	d4	base_state[4]	1	ddd	dddd	0110 0001	dddd dddd xxxX xddd
	d3	base_state[3]	1	ddd	dddd	0110 0001	dddd dddd xxxx Xddd
	d2	base_state[2]	1	ddd	dddd	0110 0001	dddd dddd xxxx xXdd
	d1	base_state[1]	1	ddd	dddd	0110 0001	dddd dddd xxxx xdXd
	d0	base_state[0]	1	ddd	dddd	0110 0001	dddd dddd xxxx xddD
5	FSM Status Word 2						
	d15	valid_int	1	ddd	dddd	0110 0010	D ddd dddd xxxx xddd
	d14	clear_valid	1	ddd	dddd	0110 0010	d Ddd dddd xxxx xddd
	d13	read_int	1	ddd	dddd	0110 0010	ddDd dddd xxxx xddd
	d12	clear_read	1	ddd	dddd	0110 0010	dddD dddd xxxx xddd
	d11	complete_cycle	1	ddd	dddd	0110 0010	dddD Dddd xxxx xddd
	d10	load_engine_data	1	ddd	dddd	0110 0010	dddD dDdd xxxx xddd
	d9	load_bounds	1	ddd	dddd	0110 0010	dddD ddDd xxxx xddd
	d8	load_counter	1	ddd	dddd	0110 0010	dddD dddD xxxx xddd
	d7	load_select	1	ddd	dddd	0110 0010	dddD dddd Xxxx xddd
	d6	reset_counter	1	ddd	dddd	0110 0010	dddD dddd x Xxx xddd
	d5	set_enable_counter	1	ddd	dddd	0110 0010	dddD dddd xx Xx xddd
	d4	reset_enable_counter	1	ddd	dddd	0110 0010	dddD dddd xxx X xddd
	d3	reset_latches	1	ddd	dddd	0110 0010	dddD dddd xxxx Xddd
	d2	zero_comp	1	ddd	dddd	0110 0010	dddD dddd xxxx xXdd
	d1	mask_zero	1	ddd	dddd	0110 0010	dddD dddd xxxx xdXd
	d0	zero	1	ddd	dddd	0110 0010	dddD dddd xxxx xddD
6	FSM Status Word 3	d15:d0	error_word	1	ddd	dddd	0110 0011
7	FSM Status Word 4	d15:d0	counter(W0[3]) LSB	1	ddd	dddd	0110 0100 input value to first SHA2_64 block
8	FSM Status Word 5	d15:d0	counter(W0[3]) MSB outh LSB, second	1	ddd	dddd	0110 0101 input value to first SHA2_64 block
9	FSM Status Word 6	d15:d0	Slice outh MSB, second	1	ddd	dddd	0110 0110 H value out of 2nd SHA2_64 block
10	FSM Status Word 7	d15:d0	Slice	1	ddd	dddd	0110 0111 H value out of 2nd SHA2_64 block
11	Reserve for chip Test via SPI bus			1	ddd	dddd	0110 1000
12	Reserve for chip Test via SPI bus			1	ddd	dddd	0110 1001
13	Reserve for chip Test via SPI bus			1	ddd	dddd	0110 1010
14	Reserve for chip Test via SPI bus			1	ddd	dddd	0110 1011
15	Reserve for chip Test via SPI bus			1	ddd	dddd	0110 1100
16	Reserve for chip Test via SPI bus			1	ddd	dddd	0110 1101
17	Reserve for chip Test via SPI bus			1	ddd	dddd	0110 1110
18	Reserve for chip Test via SPI bus			1	ddd	dddd	0110 1111

19	FIFO data				1	ddd	dddd	1000 0000	FIFO 0 <31:0> lower 16bits
20	FIFO data				1	ddd	dddd	1000 0001	FIFO 0 <31:0> upper 16bits
21	FIFO data				1	ddd	dddd	1000 0010	FIFO 1 <31:0> lower 16bits
22	FIFO data				1	ddd	dddd	1000 0011	FIFO 1 <31:0> upper 16bits
23	FIFO data				1	ddd	dddd	1000 0100	FIFO 2 <31:0> lower 16bits
24	FIFO data				1	ddd	dddd	1000 0101	FIFO 2 <31:0> upper 16bits
25	FIFO data				1	ddd	dddd	1000 0110	FIFO 3 <31:0> lower 16bits
26	FIFO data				1	ddd	dddd	1000 0111	FIFO 3 <31:0> upper 16bits
27	FIFO data				1	ddd	dddd	1000 1000	FIFO 4 <31:0> lower 16bits
28	FIFO data				1	ddd	dddd	1000 1001	FIFO 4 <31:0> upper 16bits
29	FIFO data				1	ddd	dddd	1000 1010	FIFO 5 <31:0> lower 16bits
30	FIFO data				1	ddd	dddd	1000 1011	FIFO 5 <31:0> upper 16bits
31	FIFO data				1	ddd	dddd	1000 1100	FIFO 6 <31:0> lower 16bits
32	FIFO data				1	ddd	dddd	1000 1101	FIFO 6 <31:0> upper 16bits
33	FIFO data				1	ddd	dddd	1000 1110	FIFO 7 <31:0> lower 16bits
34	FIFO data				1	ddd	dddd	1000 1111	FIFO 7 <31:0> upper 16bits
35	Global Query -- all listen query command, special read for reporting the highest prior done engine. Highest is farthest from MCU							0000 0000 0 ddd dddd present this register on the SPI bus (with the hard wired address) ONLY if this engine has done in=0, dout out =1	

13. Appendix B

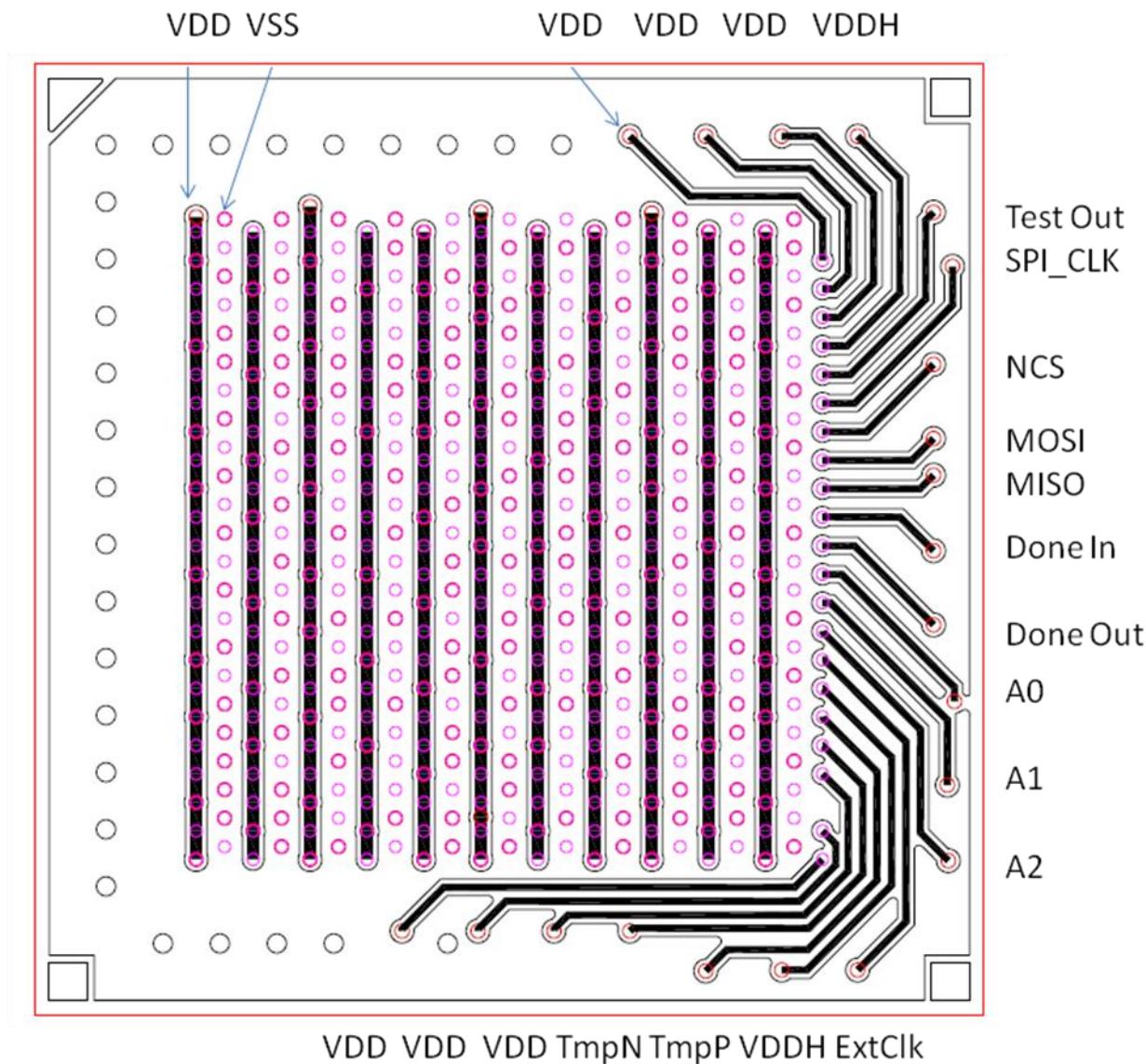
BGA Packaging – top level

The top level of the die has bump connections as shown in the picture below.
 The Logo location for reference is the bottom left hand corner of the die





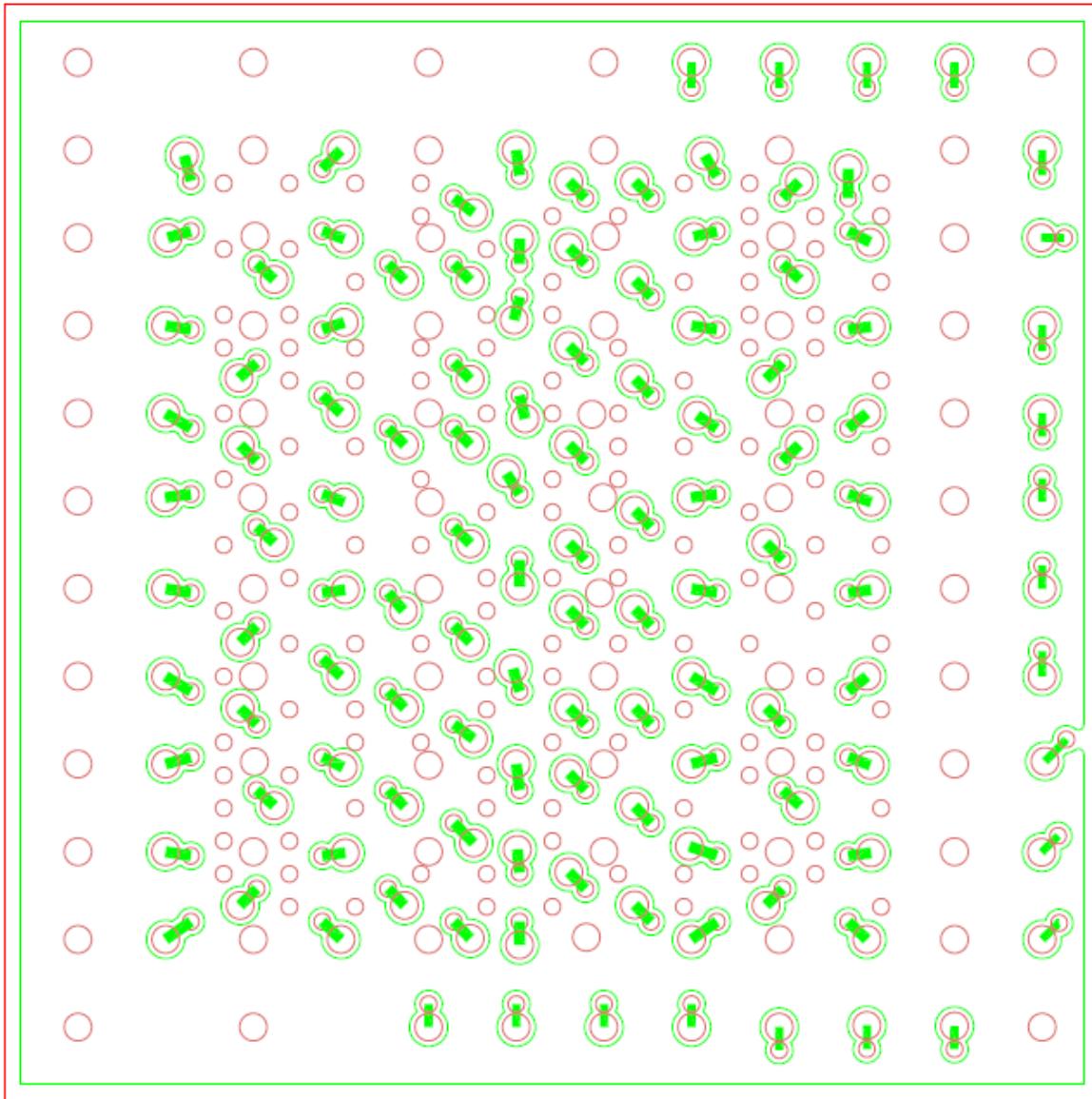
In order for the die to be assembled with the BGA package the die is flipped and the top side bump pattern is mounted to the top level of the BGA packaging assembly below. The die bumps then make contact with the 23 x 23 matrix of connections in the center of the package and the logo is now in the bottom left hand corner as a result of the flipping action





BUTTERFLYLABS

65nm Chip Specifications
BFL SHA2 ASIC

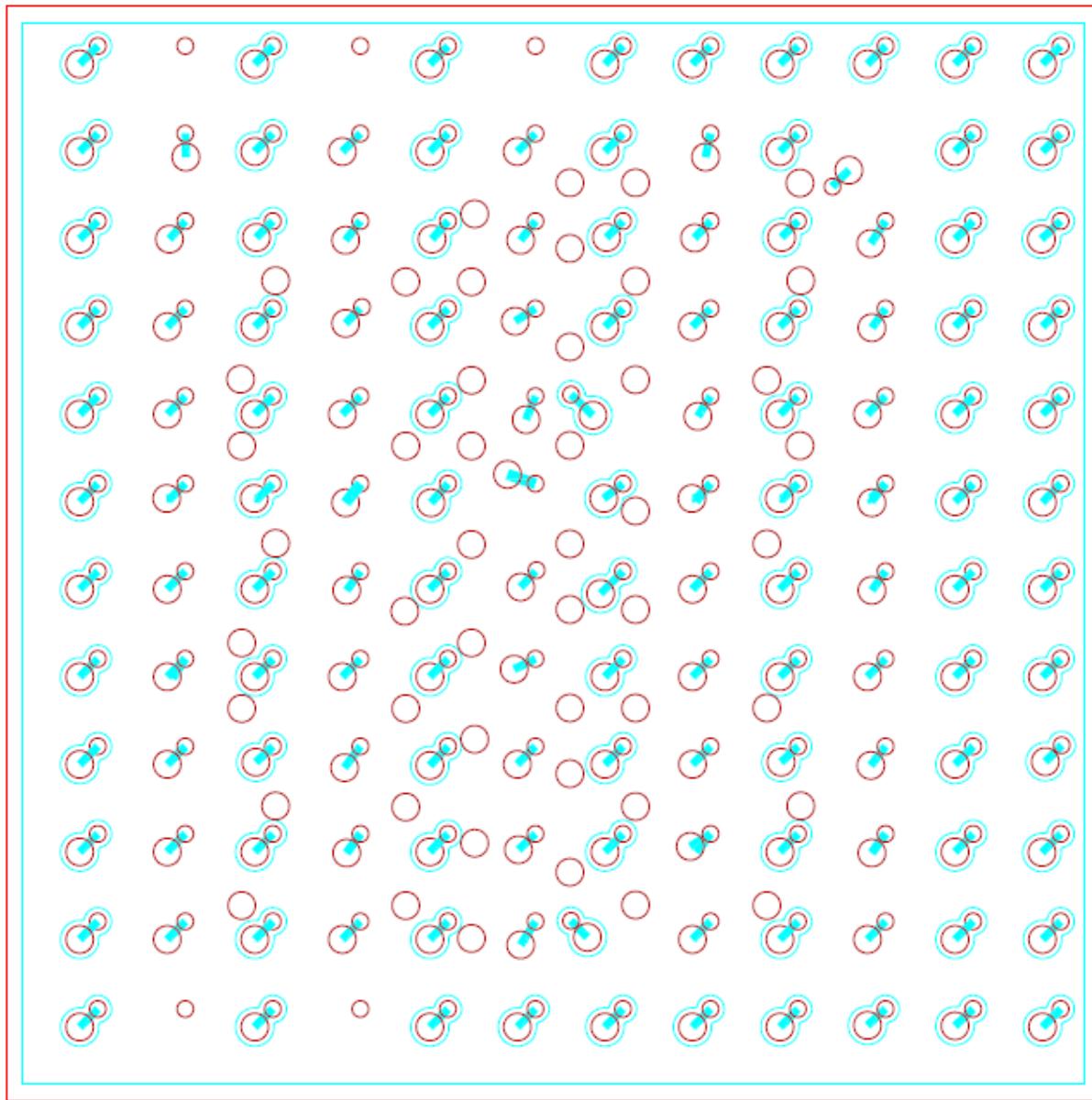




BUTTERFLYLABS

65nm Chip Specifications

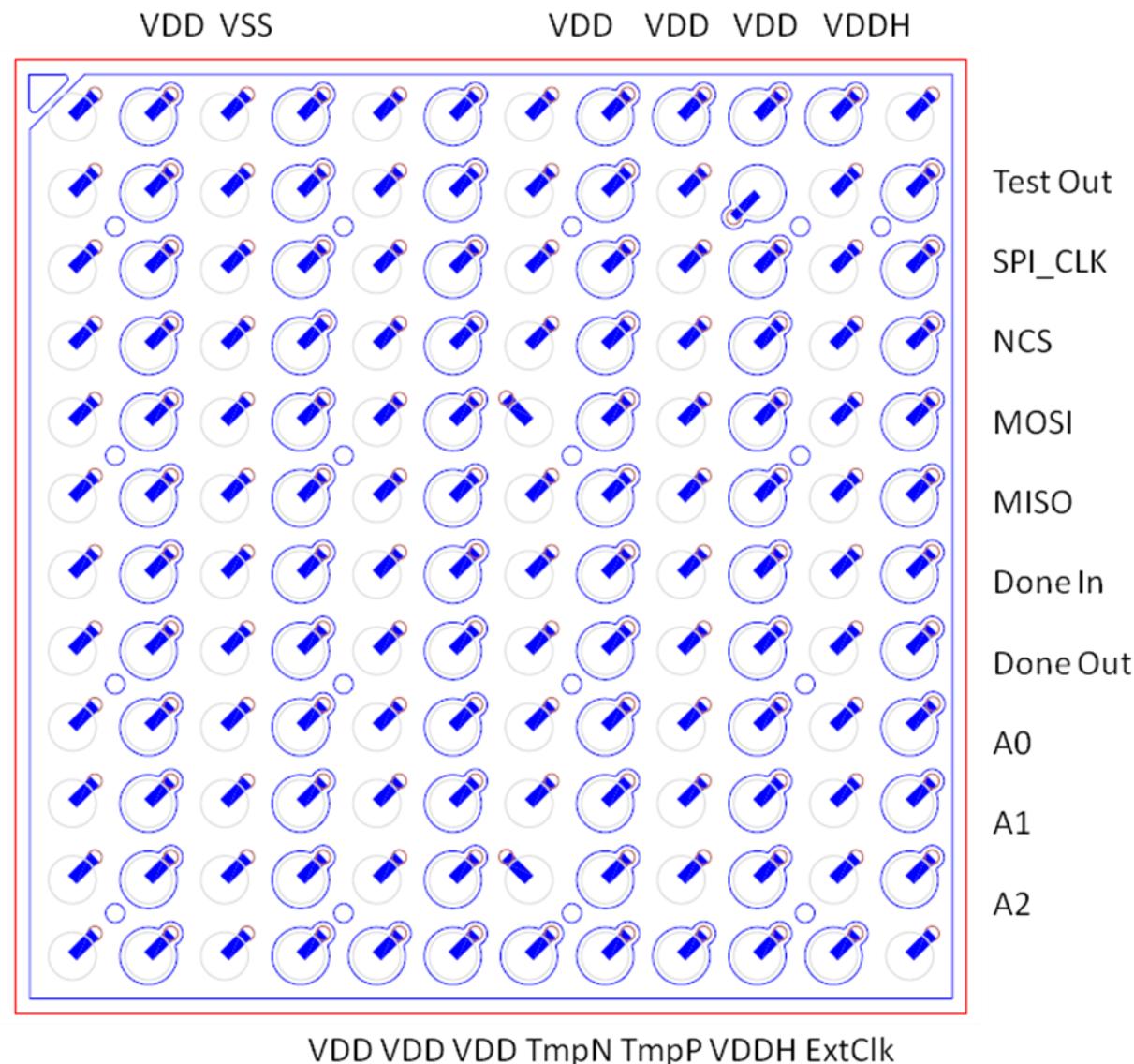
BFL SHA2 ASIC





The final package pinout as a result of the internal circuit board connectivity is shown below. In this drawing pin A1 of the package is the top left corner, all four corner pin locations are 'no connects' and the remaining pins are marked as shown.

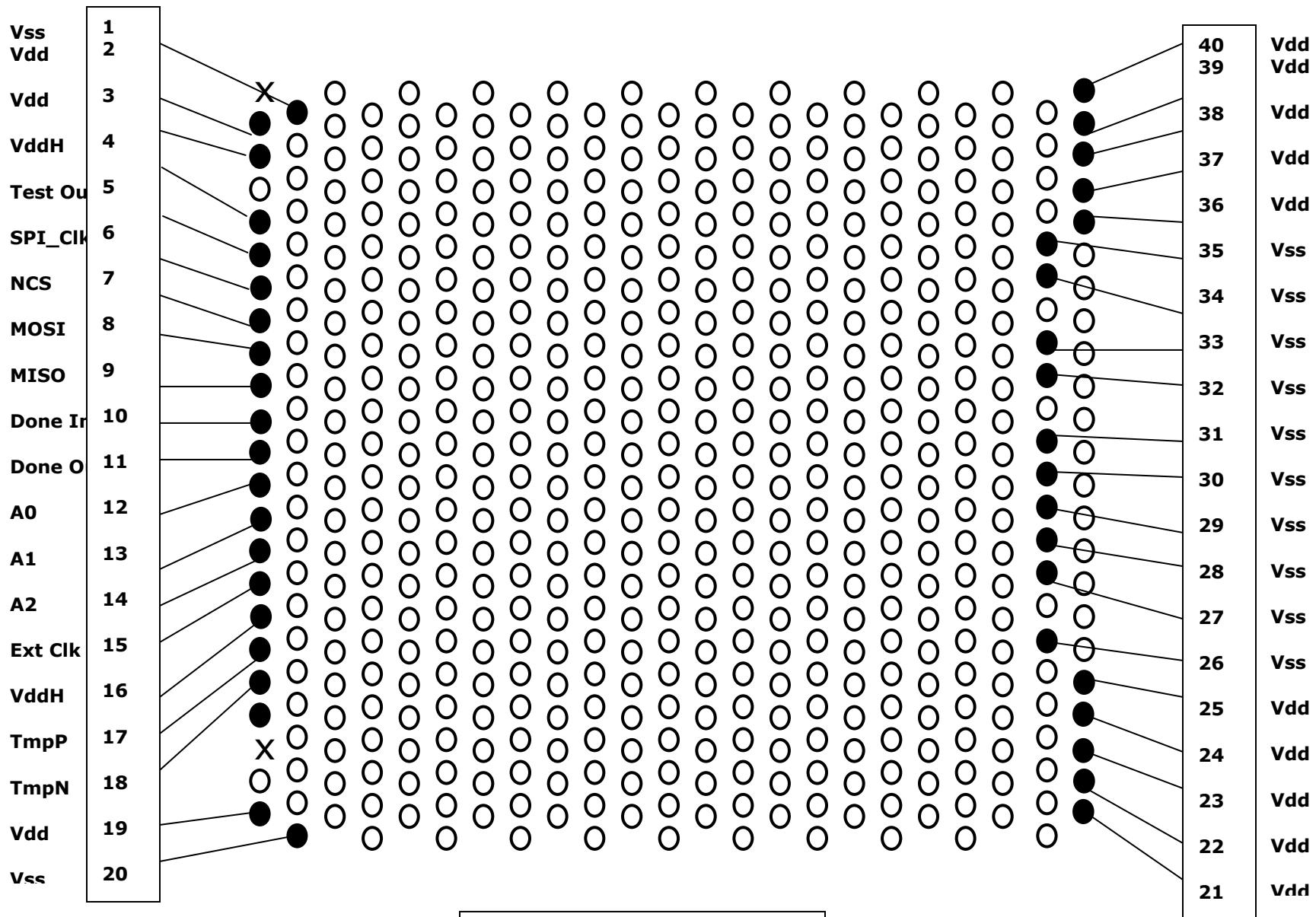
View from die side of package



40 Pin DIP Assembly

The die is also being mounted in a 40 pin DIP package for a quick assembly for expedited test. In this assembly the die is not flipped (as for the BGA) and the designated wire bond sites and corresponding pin signal names are shown in the drawing on the next page.

In this arrangement the final DIP assembly will have the logo location in the bottom right hand corner



40 Pin DIP Package



14. Appendix C

BFL Startup and brief Operation description

1. Set external clock source to be disabled – clock held low
2. Power on chip test
3. Keep NCS high and provide at least one SPI clock (+ve transition). This will reset the SPI state machine. (Normal SPI mode for Read or Write is NCS low for all 32 clocks of address and data)
3. Select external clock mode – d15 Write Register 1 (instance 0)
4. Set Reset bit (d12 Write Register 1) for all 16 instances
5. Enable clock out (d0-d15 Write Register 8) for all 16 instances
6. Provide external Hash Engine clocks to clock in Reset to all instances (should only require one external clock for this).

At this point all of the HE instances should be Reset – you can check by reading Read Register 4 - FSM Status Word 1 – all state bits should be set high.

If only one engine is being tested disable clocks for instances not being tested (d0-d15 Write Register 7).

Write vector(s) into the registers for the instance under test. The only sequence here is WriteRegistersValid gets set only after all other registers have been written. When WriteRegistersValid is written this starts the computation of the Hash Engine – so all of the other registers need to be written before this is set.

Once WriteRegistersValid is written Busy – Read Register1 d1 should go high for as long as the computation is ongoing.



The Done signal should indicate when the computation has been completed and at this point data is written to the FIFO – before this there will be nothing written to the FIFO.

Once the data has been read from the FIFO (Read Registers 19-34) and (Read Status Register) – the number of FIFOs that have valid data – ReadCompleted d10 Write Register should be written to clear the data from the FIFOs.

When using the internal oscillator Write Register 7 allows incremental adjustment of the oscillator frequency (by switching in various delay cell combinations) and provides approx 30% slow down and 10% speed up from the nominal set point of HHHHHHHHHH-LHLHLH

Changing the HH on the left to LH will increase the clock, changing the LH to HH will slow it. Changing everything to LL will slow it even more. There is no order to the register except that the data is in pairs LL is slow, LH is fast, and HL and HH is nominal.