Black Arrow Minion 99s256 Datasheet

Black Arrow Software
Support (at) blackarrowsoftware.com

REVISION HISTORY

REV	REVISION HISTORY	AUTHOR	DATE
1.0	Create	Alex, Cristina, Nobel	2013-11-8
2.0	Change temp_ctrl and freq_ctrl definition	Nobel	2013-12-19
2.1	Add chip test in the application information	Nobel	2013-12-20
3.0	Change core number to 99, change chip_signature, add idle_cnt register.	Nobel	2013-12-27
3.1	Added package information	Alex	2014-04-11
4.0	Changed the core number to 99.	Nobel	2014-5-26

Contents

1. INTRODUCTION 4
2. KEY FEATURE 4
2.1 SHA-256 4
2.2 SPI INTERFACE 5
2.3 SHA MANAGER 5
3. PACKAGE 6
3.1 Drawings 6
3.2 Pin map 7
3.3 PIN DESCRIPTION 8
4. REGISTER DESCRIPTION 9
4.1 SYSTEM REGISTERS(basic address=00h) 10
4.1.1 Chip signature register(offset address = 00h) 10
4.1.2 chip status register(offset address = 01h) 11
4.1.3 Led SPI register(offset address = 02h) 11
4.1.4 Temperature control register(offset address = 03h) 11
4.1.5 Frequency control register(offset address = 04h) 11
4.1.6 Led get nonce time register(offset address = 05h) 12
4.1.7 Miscellaneous control register(offset address = 06h) 12
4.1.8 Software reset register(offset address = 07h) 13
4.1.9 System interrupt enable register(offset address = 08h) 13
4.1.10 System interrupt clear register(offset address = 09h) 13
4.1.11 System interrupt register(offset address = 0ah) 14
4.1.12 Two FIFO status register(offset address = 0bh) 14
4.1.13 Command queue interrupt trig register(offset address = 0ch) 14
4.1.14 Result buffer interrupt trig register(offset address = 0dh) 15
4.1.15 Idle counter register(offset address = 0eh) 15
4.2 CORE REGISTERS(basic address=10h) 15
4.2.1 core enable low register(offset address = 00h) 15
4.2.2 core enable high register(offset address = 01h) 15
4.2.3 core enable high register(offset address = 02h) 15
4.2.4 core enable high register(offset address = 03h) 15
4.2.5 core idle low register(offset address = 04h) 16
4.2.6 core idle high register(offset address = 05h) 16
4.2.7 core idle high register(offset address = 06h) 16
4.2.8 core idle high register(offset address = 07h) 16
4.3 RESULT BUF ACCESS REGISTERS(basic address=20h) 16

4.3.1 Result buffer register(offset address = 00h) 16
4.3.2 Result buffer memory register(offset address = 01h) 17
4.4 COMMAND QUEUE ACCESS REGISTERS(basic address=30h) 17
4.4.1 Command queue write register(offset address = 00h) 17
4.4.2 Command queue read register(offset address = 01h) 17
4.5 NONCE REGISTERS(basic address=70h) 18
4.5.1 Start nonce register(offset address = 00h) 18
4.5.2 Increment register(offset address = 01h) 18
5. APPLICATION INFORMATION 18
5.1 Application diagram 18
5.2 Chip test 19
5.3 Initialization 19
5.4 Assign task 20
5.5 Result read back 20
5.6 Interrupt process 20
5.7 Reset control 21
5.8 LED control 21
6. ELECTIRICAL SPECIFICATION and TIMING 21
6.1 DC Characteristics 21
6.2 AC Timing Characteristics 21

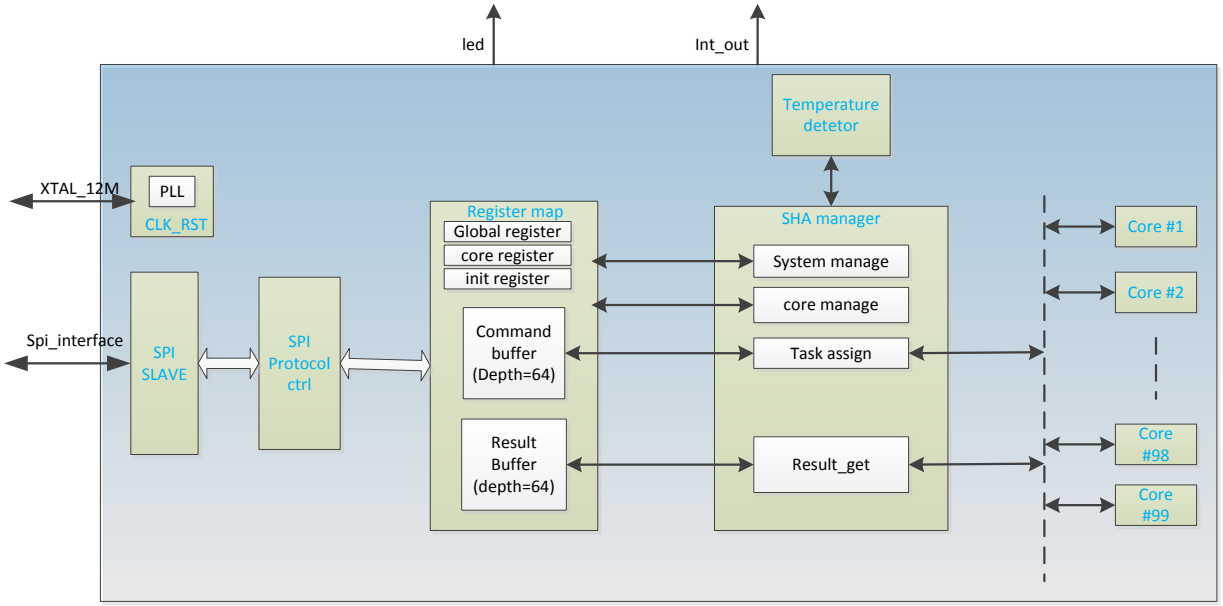
1. INTRODUCTION

This user guide describes the interfaces for the BLACKARROW MINION ASIC, and for board assemblies and related products produced by BLACKARROW. The intended audience is

- Driver developers, who wish to write device drivers for mining applications
- Third party OEM's, who are deploying BLACKARROW MINION ASICs or IP in their own hardware, and must write interface firmware or drivers associated with their value added products
- Technical users who wish to know more about the mining products they have purchased
- Any users who simply wishes to know more about BLACKARROW products

MINION is high performance, low power ASIC solution for BTC miner machine. The chip contains 64 cores with SHA-256 calculation for the nonce finding, and is easy to be expanded with other chips through sharing SPI interface.

Figure 1. System diagram



2. KEY FEATURE

2.1 SHA-256

- 1.0Ghz@TT_0p85v_25, 99Ghash/s

- 1.2Ghz@TT_0p85V_25, 118.8Ghash/s, for overclock optional.
- 99 cores embedded
- Sha-256 clock will turn off when it doesn't work for the saving power consumption
- Initial state configurable, size optimized SHA-256 core

2.2 SPI INTERFACE

- Slave mode and 0/3 timing option mode supported
- Both MSBIT first and LSBIT first is supported
- Run up to 50Mhz speed
- Continuously transmitting supported

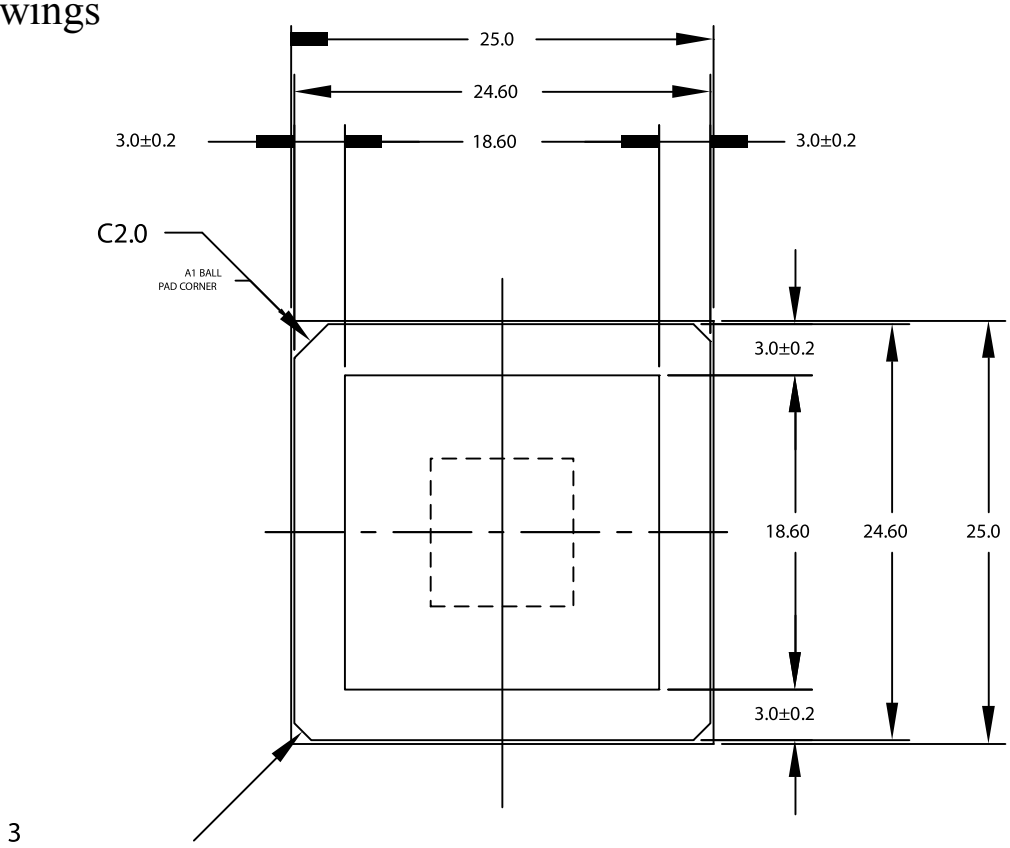
2.3 SHA MANAGER

- Read golden nonce from each core, ensure no golden nonce lost
- Easy to read golden nonce, it can read all result in one time SPI operation
- Threshold configurable temperature detector is supported
  - Threshold high: when it detects threshold high, manager will send interrupt and clock gating all cores automatically
  - Threshold low: resume work automatically
- Interrupt generated in below case
  - Command queue has more than trigger level space
  - Command queue overrun
  - Result buffer has more than trigger level
  - Result buffer overrun
  - Temperature reaches threshold high/low
- Update system/core registers based on system/cores status
- Update LED indictor status

3. PACKAGE

3.1 Drawings

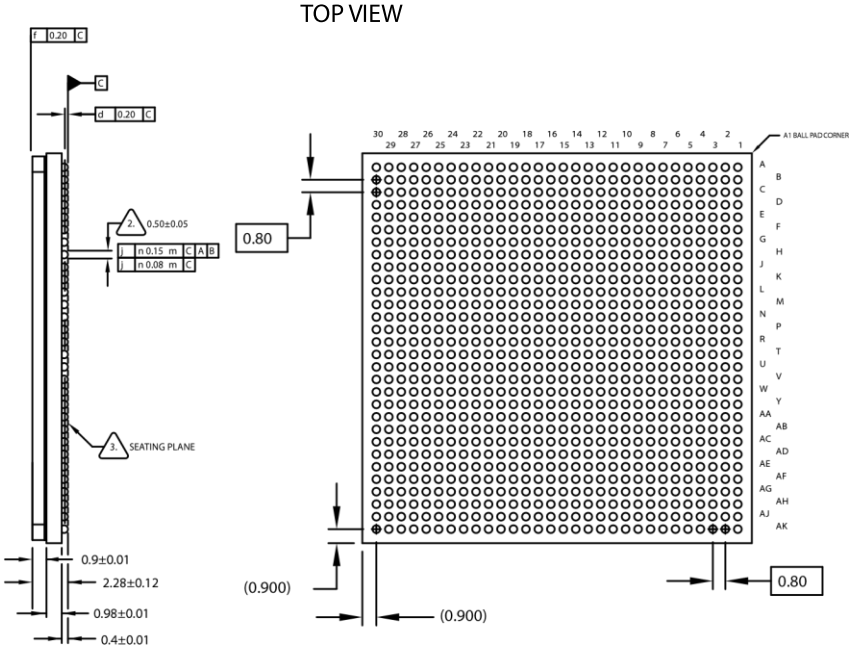
FCBGA with Stiffener



3

1.0

TOP VIEW

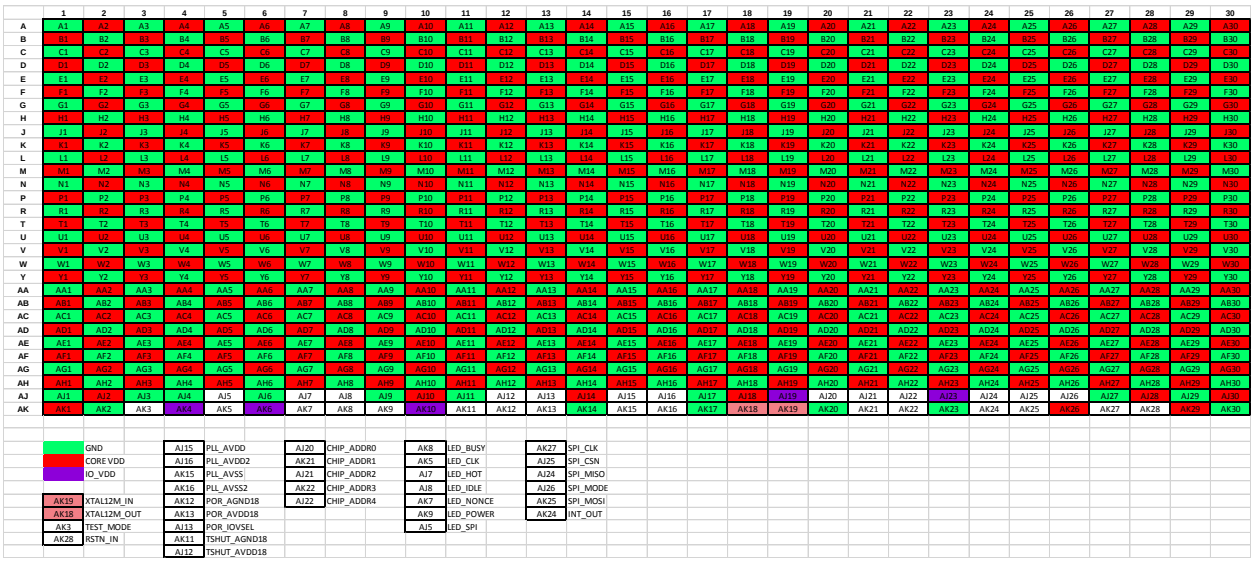


SIDE VIEW

BOTTOM VIEW
900 SOLDER BALLS

3.2 Pin map

Figure 2. Pin map



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
A	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23	A24	A25	A26	A27	A28	A29	A30
B	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17	B18	B19	B20	B21	B22	B23	B24	B25	B26	B27	B28	B29	B30
C	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	C29	C30
D	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28	D29	D30
E	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17	E18	E19	E20	E21	E22	E23	E24	E25	E26	E27	E28	E29	E30
F	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25	F26	F27	F28	F29	F30
G	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	G24	G25	G26	G27	G28	G29	G30
H	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17	H18	H19	H20	H21	H22	H23	H24	H25	H26	H27	H28	H29	H30
J	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19	J20	J21	J22	J23	J24	J25	J26	J27	J28	J29	J30
K	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11	K12	K13	K14	K15	K16	K17	K18	K19	K20	K21	K22	K23	K24	K25	K26	K27	K28	K29	K30
L	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	L16	L17	L18	L19	L20	L21	L22	L23	L24	L25	L26	L27	L28	L29	L30
M	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20	M21	M22	M23	M24	M25	M26	M27	M28	M29	M30
N	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14	N15	N16	N17	N18	N19	N20	N21	N22	N23	N24	N25	N26	N27	N28	N29	N30
P	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30
R	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	R26	R27	R28	R29	R30
T	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25	T26	T27	T28	T29	T30
U	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U15	U16	U17	U18	U19	U20	U21	U22	U23	U24	U25	U26	U27	U28	U29	U30
V	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	V29	V30
W	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	W18	W19	W20	W21	W22	W23	W24	W25	W26	W27	W28	W29	W30
Y	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12	Y13	Y14	Y15	Y16	Y17	Y18	Y19	Y20	Y21	Y22	Y23	Y24	Y25	Y26	Y27	Y28	Y29	Y30
AA	AA1	AA2	AA3	AA4	AA5	AA6	AA7	AA8	AA9	AA10	AA11	AA12	AA13	AA14	AA15	AA16	AA17	AA18	AA19	AA20	AA21	AA22	AA23	AA24	AA25	AA26	AA27	AA28	AA29	AA30
AB	AB1	AB2	AB3	AB4	AB5	AB6	AB7	AB8	AB9	AB10	AB11	AB12	AB13	AB14	AB15	AB16	AB17	AB18	AB19	AB20	AB21	AB22	AB23	AB24	AB25	AB26	AB27	AB28	AB29	AB30
AC	AC1	AC2	AC3	AC4	AC5	AC6	AC7	AC8	AC9	AC10	AC11	AC12	AC13	AC14	AC15	AC16	AC17	AC18	AC19	AC20	AC21	AC22	AC23	AC24	AC25	AC26	AC27	AC28	AC29	AC30
AD	AD1	AD2	AD3	AD4	AD5	AD6	AD7	AD8	AD9	AD10	AD11	AD12	AD13	AD14	AD15	AD16	AD17	AD18	AD19	AD20	AD21	AD22	AD23	AD24	AD25	AD26	AD27	AD28	AD29	AD30
AE	AE1	AE2	AE3	AE4	AE5	AE6	AE7	AE8	AE9	AE10	AE11	AE12	AE13	AE14	AE15	AE16	AE17	AE18	AE19	AE20	AE21	AE22	AE23	AE24	AE25	AE26	AE27	AE28	AE29	AE30
AF	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10	AF11	AF12	AF13	AF14	AF15	AF16	AF17	AF18	AF19	AF20	AF21	AF22	AF23	AF24	AF25	AF26	AF27	AF28	AF29	AF30
AG	AG1	AG2	AG3	AG4	AG5	AG6	AG7	AG8	AG9	AG10	AG11	AG12	AG13	AG14	AG15	AG16	AG17	AG18	AG19	AG20	AG21	AG22	AG23	AG24	AG25	AG26	AG27	AG28	AG29	AG30
AH	AH1	AH2	AH3	AH4	AH5	AH6	AH7	AH8	AH9	AH10	AH11	AH12	AH13	AH14	AH15	AH16	AH17	AH18	AH19	AH20	AH21	AH22	AH23	AH24	AH25	AH26	AH27	AH28	AH29	AH30
AJ	AJ1	AJ2	AJ3	AJ4	AJ5	AJ6	AJ7	AJ8	AJ9	AJ10	AJ11	AJ12	AJ13	AJ14	AJ15	AJ16	AJ17	AJ18	AJ19	AJ20	AJ21	AJ22	AJ23	AJ24	AJ25	AJ26	AJ27	AJ28	AJ29	AJ30
AK	AK1	AK2	AK3	AK4	AK5	AK6	AK7	AK8	AK9	AK10	AK11	AK12	AK13	AK14	AK15	AK16	AK17	AK18	AK19	AK20	AK21	AK22	AK23	AK24	AK25	AK26	AK27	AK28	AK29	AK30

	GND	AJ15	PLL_AVDD	AJ20	CHIP_ADDR0	AK8	LED_BUSY	AK27	SPI_CLK	
	CORE VDD	AJ16	PLL_AVDD2	AK21	CHIP_ADDR1	AK5	LED_CLK	AJ25	SPI_CSN	
	IO_VDD	AK15	PLL_AVSS	AJ21	CHIP_ADDR2	AJ7	LED_HOT	AJ24	SPI_MISO	
		AK16	PLL_AVSS2	AK22	CHIP_ADDR3	AJ8	LED_IDLE	AJ26	SPI_MODE	
AK19	XTAL12M_IN	AK12	POR_AGND18	AJ22	CHIP_ADDR4	AK7	LED_NONCE	AK25	SPI_MOSI	
AK18	XTAL12M_OUT	AK13	POR_AVDD18			AK9	LED_POWER	AK24	INT_OUT	
AK3	TEST_MODE	AJ13	POR_IOVSEL			AJ5	LED_SPI			
AK28	RSTN_IN	AK11	TSHUT_AGND18							
		AJ12	TSHUT_AVDD18							

3.3 PIN DESCRIPTION

Table 1. PIN LIST

symbol	Type	Pin NO.	Description
• Global signal			
CHIP ADDR4	DI		ID of ASIC
CHIP ADDR3	DI		
CHIP ADDR2	DI		
CHIP ADDR1	DI		
CHIP ADDR0	DI		
INT_OUT	DO		Interrupt output
PLL_CLK	DI		Global clock
RSTN_IN	DI		Reset signal
LED_POWER	DO		Led power indicator 1: powered 0: default chip off
LED_SPI	DO		LED indicator, which is controlled by SPI
LED_BUSY	DO		Chip busy indicator
LED_IDLE	DO		Core idle chip busy indicator: 1: Busy 0: Idle
LED_NONCE	DO		nonce get indicator,,: 1: Get nonce, stay time configurable 0:no nonce get currently
LED_HOT	DO		Chip hot indicator 1: high threshold triggered 0: low threshold triggered
TEST_MODE	DO		Test mode 1: test mode 0: function mode
• SPI signal			
SPI_MODE	DI		SPI mode select 1: mode 3 0: mode 1
SPI_CLK	DI		SPI slave clock in
SPI_CSN	DI		SPI select signal 1: disable 0: select enable
SPI_MOSI	DI		SPI slave input data
SPI_MISO	DO		SPI slave output data
SPI_OE	DO		SPI slave output data enable
• Power pins			
TD_AVDD			Temperature detector power
TD_AVSS			Temperature detector gnd
PLL_AVDD			PLL power
PLL_AVSS			PLL GND
IO_VDD[19:0]			IO power
IO_VSS[19:0]			IO GND
CORE_VDD[63:0]			Core power
CORE_VDD[63:0]			Core gnd

4. REGISTER DESCRIPTION

Only SPI slave port can access these addressable registers. The SPI timing is shown in the Figure 6

There are two FIFO access address (CMD queue/Result Buf), these two address will remain the same during each SPI command. And other addresses will automatically add 1 after each data has been transferred.

The SPI command protocol please see below:

One whole command contains two parts: one is the header and another is data. The header contains 4 parts: chip id, read/write label, register address, and data length.

Chip id indicates the chip name we want to operate. Read/write label high indicates current SPI command is a read command and low indicates current SPI command is a write command. Register address is the chip register address. Our chip can operates SPI continuously, so data length shows how many data the master want to write or read.

Data length is measured in bytes. Data length = 4 means transmit 4 bytes (32 bits), data length = 8 means transmit 8 bytes (64 bits)

Figure 3. SPI header format

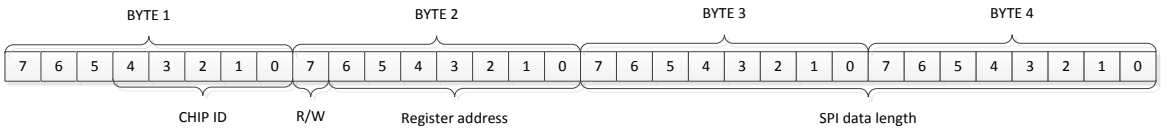


Figure 4. SPI command format

SPI slave interface supports mode 0 and mode 3, data is synchronized with SPI_CLK rise edge. Each command should finish until SPI_CSN de-assert.

SPI interface has two transmit methods: MSBIT first and LSBIT first. It is selected by bit 0 of register misc_ctrl.

Table 2. SPI CMD/DATA format(MSbit first)

(8 bits) {3 bits 0 + 5 bits CHIP ID}	3'b000, CHIP_ID[4], CHIP_ID [3], CHIP_ID [2], CHIP_ID [1], CHIP_ID [0]
(8 bits){1 bit r/w + 7 bit address}	r/w, address[6], address[5], ……, address[1], address[0]
(16 bits data length)	Len[7], Len[6],……, Len[0], Len[15], Len[14], ……., Len[8]

(8 bits) {3 bits 0 + 5 bits CHIP ID}	3'b000, CHIP_ID[4], CHIP_ID [3], CHIP_ID [2], CHIP_ID [1], CHIP_ID [0]
(32 bits data0)	data_0[7],data_0[6],...data_0[0],data_0[15],...data_0[8],data_0[23],...data_0[16], data_0[31],...data_0[24]
(32 bits data_n)	data_n[7],data_n[6],...data_n[0],data_n[15],...data_n[8],data_n[23],...data_n[16].....data_n[31],...data_n[24]

Table 3. SPI CMD/DATA format(LSbit first)

(8 bits) {3 bits 0 + 5 bits CHIP ID}	CHIP_ID [0], CHIP_ID [1], CHIP_ID [2], CHIP_ID [3], CHIP_ID [4] ,3'b000,
(8 bits){1 bit r/w + 7 bit address}	address[0], address[1], ……, address[5], address[6], r/w
(16 bits data length)	Len[0], Len[1],……, Len[7], Len[8], Len[9], ……., Len[15]
(32 bits data0)	data_0[0],data_0[1],...data_0[8],data_0[9],...data_0[15],data_0[16],...data_0[23], data_0[24],...data_0[31]
(32 bits datan)	data_n[0],data_n[1],...data_n[7],data_n[8],...data_n[15],data_n[16],...data_n[23], data_n[24],...data_n[31]

Table 4. registers overview

Register Category	**Base Address**
System registers	00h
core registers	10h
result buffer access registers	20h
CMD queue access registers	30h
Nonce registers	70h

4.1 SYSTEM REGISTERS(basic address=00h)

4.1.1 Chip signature register(offset address = 00h)

Table 5. Chip signature register map

Name	Default value	Label	Bit	R/W	Description
Chip_signature	32'hb1ac8a44	Chip_signature	31:0	RO	Chip signature

4.1.2 chip status register(offset address = 01h)

Table 6. Chip status register map

Name	Default value	Label	Bit	R/W	Description
Chip_status	32'h006303e8	Temp_sta	31:24	RO	temperature detector current value
		Core_number	23:16		number of cores in this chip
		Design_frequency	15:0		design frequency of this chip in unit of Mhz

4.1.3 Led SPI register(offset address = 02h)

Table 7. Led SPI register map

Name	Default value	Label	Bit	R/W	Description
Led_spi_ctrl	32'h00000000	Reserved	31:16	R/W	Reserved
		Led_spi_ctrl	15:0		Write 16'ha5a5 to this register will light the LED_SPI led signal

4.1.4 Temperature control register(offset address = 03h)

Table 8. Temperature control register map

Name	Default value	Label	Bit	R/W	Description
Temp_ctrl	32'h00010004	Temp_sample_r	31:8	R/W	Temp_sample_r Temperate sensor sample rate Fsample=Fsys/(temp_sample_r+1)
		Reserved	7:6		Reserved
		Temp_det_byp	5		1:bypass temperature detector 0:temperature detector enable
		Temp_hys_window	4		Temperature hysteresis window select 1:40℃ hysteresis window 0:20℃ hysteresis window
		Temp_thres	3:0		temperature threshold

4.1.5 Frequency control register(offset address = 04h)

Table 9. Frequency control register map

Name	Default value	Label	Bit	R/W	Description
freq_ctrl	32'h0030002c	Reserved	31:24	R/W	Reserved
		Sys_div	23:21		Sys_clk division factor Sys_clk=core_clk/(sys_div+1)
		Pll_div_2	20		1- core_clk=pll_clk/2 0- core_clk=pll_clk
		Reserved	19		Reserved

		Pll_byp	18		1-pll bypass 0-clock output from pll Fpll=Fref*(pll_dn+1)/((pll_dm+1)*(pll_dp+1))
		Pll_cont	17:16		pll output clock frequency range select
		Reserved	15		Reserved
		Pll_dp	14:12		dp value for pll
		Pll_dm	11:8		dm value for pll
		Pll_dn	7:0		dn value for pll

4.1.6 Led get nonce time register(offset address = 05h)

Table 10. Led get nonce time register map

Name	Default value	Label	Bit	R/W	Description
Led_get_nonce_time	32'h000000ff	Led_get_nonce_time	31:0	R/W	LED remaining time = Tcycle*led_get_nonce_time

4.1.7 Miscellaneous control register(offset address = 06h)

Table 11. Miscellaneous control register map

Name	Default value	Label	Bit	R/W	Description
Misc_ctrl	32'h00000000	Reserved	31:6	R/W	Reserved
		Idle_cnt_mode	5		1-each 2^20 sys_clock idle_cnt +1 0-each 2^10 sys_clock idle_cnt+1
		Cores_pause	4		1-enable pause cores function 0-disable pause cores function
		Fetch_disable	3		This bit disable SHA manager fetch CMD from command queue When enable this bit, core will pause after complete current task and not fetch new task from command buffer 1-disable fetch command 0-enable fetch command
		No_nonce_flag_en	2		When enabling this bit, cores will also report into the output result queue the tasks that have no valid nonce 1-enable no nonce report to result_buf 0-disable no nonce report to result buf

		Sw_test	1		1-software test mode enable 0-normal function mode In software test mode, SHA manager send same work to all enabled cores, but task id sent to the core will be the core id. core0 has task id 0, core1 has task id 1... all works from start_nonce to start_nonce+interv
		Spi_endian	0		1-LSbits first 0-MSbits first

4.1.8 Software reset register(offset address = 07h)

Table 12. Software reset register map

Name	Default value	Label	Bit	R/W	Description
Sw_rstn	32'h0000001f	Sys_sw_rstn	31:16	R/W	Write 0xf5a5 will reset whole chip
		Reserved	15:5		Reserved
		Sham_sw_rstn	4		SHA manager software reset, write 0 to reset the module
		Spi_sw_rstn	3		Spi slave module software reset, write 0 to reset the module
		Cmd_que_sw_rstn	2		flush all CMD in queue, write 0 to reset the module
		Result_buf_sw_rstn	1		flush all golden nonce in result buf, write 0 to reset the module
		All_cores_sw_rstn	0		Reset all cores, write 0 to reset the module

4.1.9 System interrupt enable register(offset address = 08h)

Table 13. System interrupt enable register map

Name	Default value	Label	Bit	R/W	Description
Sys_int_en	32'h00000000	Reserved	31:6	R/W	Reserved
		Sys_int_en	5:0		enable interrupt of related bit in sys_int set high to enable it.

4.1.10 System interrupt clear register(offset address = 09h)

Table 14. System interrupt clear register map

Name	Default value	Label	Bit	R/W	Description
Sys_int_clr	32'h00000000	Reserved	31:6	R/W	Reserved

		Sys_int_clr	5:0		clear interrupt of related bit in sys_int Automatically go back to default value after set to 1 Set 1 to clear interrupt

4.1.11 System interrupt register(offset address = 0ah)

Table 15. System interrupt register map

Name	Default value	Label	Bit	R/W	Description
Sys_int	32'h00000000	Reserved	31:6	RO	Reserved
		temp_high_int	5		temperature go up to threshold high
		temp_low_int	4		temperature go down to threshold low
		cmd_que_overrun_int	3		SPI still send cmd to queue when cmd queue full
		cmd_que_vld_int	2		cmd queue valid,this interrupt will be triggered when cmd queue has more spaces than "cmd_que_int_trig"
		result_buf_overrun_int	1		cores still send nonce when result buf full
		result_buf_vld_int	0		chip get_nonce interrupt. This interrupt will be triggered when result buf has more items than "result_buf_int_trig"

4.1.12 Two FIFO status register(offset address = 0bh)

Table 16. Two FIFO status register map

Name	Default value	Label	Bit	R/W	Description
Two_fifo_status	32'h00000000	Reserved	31:16	RO	Reserved
		cmd_que_sta	15:8		how many command items in command queue
		result_buf_sta	7:0		how many result items in result buf now

4.1.13 Command queue interrupt trig register(offset address = 0ch)

Table 17. Command queue interrupt trig register map

Name	Default value	Label	Bit	R/W	Description
Cmd_que_int_trig	32'h00000001	Reserved	31:7	R/W	Reserved
		cmd_que_int_trig	6:0		when cmd queue has more spaces than this trigger

					level, “cmd_que_vld_int” interrupt will be triggered

4.1.14 Result buffer interrupt trig register(offset address = 0dh)

Table 18. Result buffer interrupt trig register map

Name	Default value	Label	Bit	R/W	Description
Result_buf_int_trig	32’h00000001	Reserved	31:7	R/W	Reserved
		Result_buf_int_trig	6:0		The trigger level for result buf, when result buf has items more than this trigger level, “result_buf_vld_int” interrupt will be triggered

4.1.15 Idle counter register(offset address = 0eh)

Table 19. Idle counter trig register map

Name	Default value	Label	Bit	R/W	Description
Idle_cnt	32’h00000000	Idle_cnt	31:0	R/W	This register will auto increase 1 when enabled core stopped by abnormal reason, including temperature high or result_buf_full

4.2 CORE REGISTERS(basic address=10h)

4.2.1 core enable low register(offset address = 00h)

Table 20. Core enable low register map

Name	Default value	Label	Bit	R/W	Description
Core_enable_l	32’hffffffff	Core_enable	31:0	R/W	each bit related to each core(core31-0)

4.2.2 core enable high register(offset address = 01h)

Table 21. Core enable high register map

Name	Default value	Label	Bit	R/W	Description
Core_enable_h	32’hffffffff	Core_enable	31:0	R/W	each bit related to each core(core63-32)

4.2.3 core enable high register(offset address = 02h)

Table 22. Core enable high register map

Name	Default value	Label	Bit	R/W	Description
Core_enable_h2	32’hffffffff	Core_enable	31:0	R/W	each bit related to each core(core95-64)

4.2.4 core enable high register(offset address = 03h)

Table 23. Core enable high register map

Name	Default value	Label	Bit	R/W	Description
Core_enable_h3	32'h00000007	Reserved	31:3	R/W	Reserved
		Core_enable	2:0		each bit related to each core(core98-96)

4.2.5 core idle low register(offset address = 04h)

Table 24. Core idle low register map

Name	Default value	Label	Bit	R/W	Description
Core_idle_l	32'hffffffff	Core_idle	31:0	RO	indicate each core is idle or not(core31-0)

4.2.6 core idle high register(offset address = 05h)

Table 25. Core idle high register map

Name	Default value	Label	Bit	R/W	Description
Core_idle_h	32'hffffffff	Core_idle	31:0	RO	indicate each core is idle or not(core63-32)

4.2.7 core idle high register(offset address = 06h)

Table 26. Core idle high register map

Name	Default value	Label	Bit	R/W	Description
Core_idle_h2	32'hffffffff	Core_idle	31:0	RO	indicate each core is idle or not(core95-64)

4.2.8 core idle high register(offset address = 07h)

Table 27. Core idle high register map

Name	Default value	Label	Bit	R/W	Description
Core_idle_h3	32'h00000007	Reserved	31:3	RO	Reserved
		Core_idle	2:0		indicate each core is idle or not(core98-96)

4.3 RESULT BUF ACCESS REGISTERS(basic address=20h)

4.3.1 Result buffer register(offset address = 00h)

Table 28. Result buffer register map

Name	Default value	Label	Bit	R/W	Description
Result_buf_0	32'h00000000	No_nonce_flag	31	RO	1-no nonce for this task 0-valid golden nonce
		Reserved	30:29		Reserved
		Chip_id	28:24		default value input port CHIP_ADDR[4:0]
		Core_id	23:16		Core_id Core_id = 0 when no_nonce_flag =1

		Task_id_bk	15:0		the right nonce related task id
Result_buf_1	32'h00000000	Nonce_bk	31:0	RO	The right nonce get back

4.3.2 Result buffer memory register(offset address = 01h)

Table 29. Result buffer memory register map

Name	Default value	Label	Bit	R/W	Description
Result_buf_mem_0	32'h00000000	No_nonce_flag	31	RO	1-no nonce for this task 0-valid golden nonce
		Reserved	30:29		Reserved
		Chip_id	28:24		default value input port CHIP_ADDR[4:0]
		Core_id	23:16		Core_id Core_id = 0 when no_nonce_flag =1
		Task_id_bk	15:0		the right nonce related task id
Result_buf_mem_1	32'h00000000	Nonce_bk	31:0	RO	The right nonce get back

Note: result buffer register is for normal use, it return the latest nonce result and reflects the FIFO output. Result buffer memory register is the memory output, you can read all nonce in one time by operating this register. You can put the SPI length equal to the number you want to read multiply by 2. For example, you want to read 16 nonce result, you can put the SPI length equal with 32.

4.4 COMMAND QUEUE ACCESS REGISTERS(basic address=30h)

4.4.1 Command queue write register(offset address = 00h)

Table 30. Command queue write register map

Name	Default value	Label	Bit	R/W	Description
Cmd_que_0	32'hffffffff	Reserved	31:16	WO	Reserved
		Task_id_in	15:0		Task id inupt
Cmd_que_1	32'hffffffff	Mid_sta_0	31:0	WO	Midstate[255:224]
Cmd_que_2	32'hffffffff	Mid_sta_1	31:0	WO	Midstate[223:192]
Cmd_que_3	32'hffffffff	Mid_sta_2	31:0	WO	Midstate[191:160]
Cmd_que_4	32'hffffffff	Mid_sta_3	31:0	WO	Midstate[159:128]
Cmd_que_5	32'hffffffff	Mid_sta_4	31:0	WO	Midstate[127:96]
Cmd_que_6	32'hffffffff	Mid_sta_5	31:0	WO	Midstate[95:64]
Cmd_que_7	32'hffffffff	Mid_sta_6	31:0	WO	Midstate[63:32]
Cmd_que_8	32'hffffffff	Mid_sta_7	31:0	WO	Midstate[31:0]
Cmd_que_9	32'hffffffff	Hash_d_0	31:0	WO	Hash_value[95:64]
Cmd_que_10	32'hffffffff	Hash_d_1	31:0	WO	Hash_value[63:32]
Cmd_que_11	32'hffffffff	Hash_d_2	31:0	WO	Hash_value[31:0]

4.4.2 Command queue read register(offset address = 01h)

Table 31. Command queue read register map

Name	Default value	Label	Bit	R/W	Description
Cmd_que_r_0	32'hffffffff	Reserved	31:16	RO	Reserved
		Task_id_in	15:0		Task id inupt
Cmd_que_r_1	32'hffffffff	Mid_sta_0	31:0	RO	Midstate[255:224]

Cmd_que_r_2	32'hffffffff	Mid_sta_1	31:0	RO	Midstate[223:192]
Cmd_que_r_3	32'hffffffff	Mid_sta_2	31:0	RO	Midstate[191:160]
Cmd_que_r_4	32'hffffffff	Mid_sta_3	31:0	RO	Midstate[159:128]
Cmd_que_r_5	32'hffffffff	Mid_sta_4	31:0	RO	Midstate[127:96]
Cmd_que_6	32'hffffffff	Mid_sta_5	31:0	RO	Midstate[95:64]
Cmd_que_7	32'hffffffff	Mid_sta_6	31:0	RO	Midstate[63:32]
Cmd_que_8	32'hffffffff	Mid_sta_7	31:0	RO	Midstate[31:0]
Cmd_que_9	32'hffffffff	Hash_d_0	31:0	RO	Hash_value[95:64]
Cmd_que_10	32'hffffffff	Hash_d_1	31:0	RO	Hash_value[63:32]
Cmd_que_11	32'hffffffff	Hash_d_2	31:0	RO	Hash_value[31:0]

Note: These two registers are like a pair of result buffer register, the command write register is the FIFO input, and it can't read back. The command read register is the FIFO output, and it can output all data in one time SPI operation. If you want to read all data in one time, you should put the SPI length equal to the number you want to read multiply by 2.

4.5 NONCE REGISTERS(basic address=70h)

4.5.1 Start nonce register(offset address = 00h)

Table 32. Start nonce register map

Name	Default value	Label	Bit	R/W	Description
Start_nonce	32'h00000000	Start_nonce	31:0	R/W	Start nonce for the first core

4.5.2 Increment register(offset address = 01h)

Table 33. increment register map

Name	Default value	Label	Bit	R/W	Description
Interv	32'h0295fad5	Interv	31:0	R/W	Increment=interv+1 Nonce increment for each core Interv should not less than 1

5. APPLICATION INFORMATION

By setting some register operation, MINION provides BTC mining automatically. This chapter describes how it can be operated

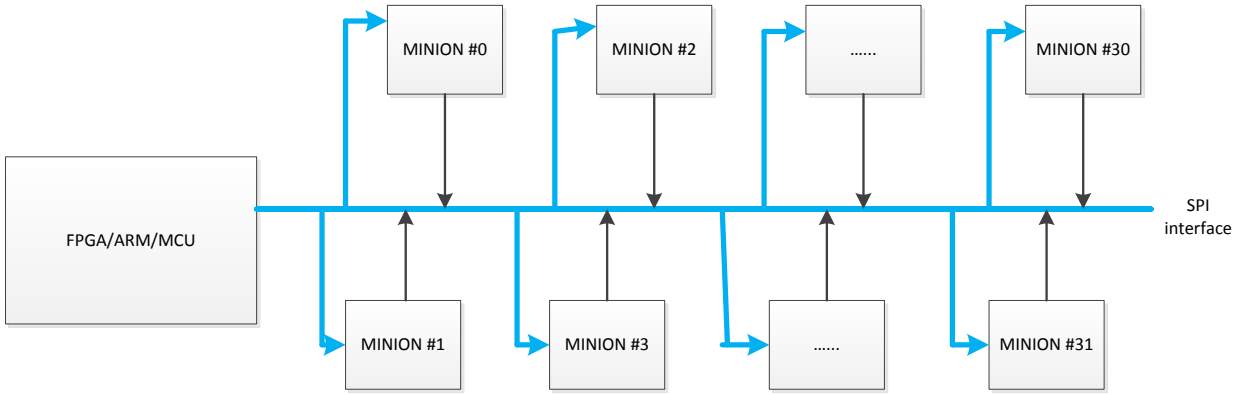
5.1 Application diagram

The chips share the same SPI BUS, and FPGA/ARM/MCU distinguish chip by chip id, for example:

When user wants to control MINION #0, it should set the CHIP ID to 0, so chips except MINION #0 will not acknowledge the SPI command.

When user wants to read back SPI command, the chip that master wants to operate will pull up the SPI_OE signal to select.

Figure 5. CHIP connection



5.2 Chip test

User should make sure the chip is good before normal work start.

- Check led
  - Led power test: user should observe the led_power_o, this led should be on once the chip is power on. If it is off, means chip has power on problem.
  - Led spi test: configure led SPI register ([15:0]=16'ha5a5) to turn on LED_SPI. It is on means that spi interface can work correctly, or off means the chip encounters some problem, user should analyze and solve it.
- Check SPI
  - User should operate the SPI to check if it is normal. It can be done by reading all default value of register or writing some register and read it.
- Soft test
  - User can check if all cores are work correctly by soft testing. User should enable soft test (configure miscellaneous control register[1]=1'b1) and set end_nonce (configure increment register[31:0]=32'hffffffff). After doing that, user can send any tasks having nonce. User can read back nonce from result buffer to check, all cores should return back the same results, if one or more cores can't return nonce or return wrong results, we can determine this core has problem.

5.3 Initialization

- Basic setting
  - Interrupt enable (configure system interrupt enable register[5:0]= 6'h3f)
  - Result interrupt threshold configuration, set 32 means it will report interrupt when it has 32 nonce in the result buffer (configure result buffer interrupt trig register[5:0] = 6'h20) or any value you want.
  - command interrupt threshold configuration, set 60 means it will report interrupt when it only has 4 nonce (64-60 =4) in the command buffer (configure command buffer interrupt trig register[5:0] = 6'h3c) or any value you want.
  - Temperature control enable (default value)
  - Core enable, enable the core which user want them to work (configure core enable low register and core enable high register), in most cases, user can use default value and enable 64 cores.
  - Nonce interval configuration (configure the start_nonce and interv).
  - for example: if the start_nonce=0x00000000, end_nonce=0xffffffff, the cores number=99, then user should configure the register: start_nonce=0x00000000, interv=0x295fad3, so we can see:

core #0 start nonce=0x0, end nonce=0x295fad3,
core #1 start nonce=0x295fad4, end nonce=0x52bf5a7
……
Core #62 start nonce=0xfad40a58, end nonce=0xfd6a052b
Core #63 start nonce=0xfd6a052c, end nonce=0xffffffff
For users, we don't need to configure every core's start nonce and end nonce and just need configure start nonce and end nonce of whole chip. Chip will calculate the start nonce and end nonce of every cores automatically

5.4 Assign task

- Data prepare
  - Send task_id, midstate and hash_value (configure command queue write register). Notes: user must send task_id, midstate and hash_value at one time, it means the spi data length should be multiples of 48
  - Master can send a serials of task to chip since chips have command memory to store it, and it will less operates time.
  - Wait for the result buffer valid interrupt

5.5 Result read back

- result read back
  - When external system monitored the result_buf_vld_int, the external system should read the nonce from the chip, and after reading, clear the interrupt.
  - MINION chip set the nonce difficulty is 0, it means at lease last 32 bits of hash value is 0 and maybe its difficulty is more than 0.
  - PC will recalculate the hash value using the nonce result, and get the correct difficulty.

5.6 Interrupt process

- Interrupt process
  - User catches interrupt by detecting int_out_o signal, which set high when there is interrupt. When users detect interrupt, we should read register sys_int, and process the interrupt. After doing that, users should clear interrupt by writing register sys_int_clr.
- Temperature high interrupt
  - When chip report temp_high_int interrupt, it means the chip temperature is too high. It should shut down the chip immediately or pause all cores work for protecting the chip, and then check what happened.
  - After processing the fault, user should start the chip and do the initialization job.
- Temperature low interrupt
  - When chip report temp_low_int interrupt, it means the chip is not work correctly, because chip temperature shouldn't be too low if it works normally. User should reset the chip or pause all cores work, and then check what happened.
  - After processing the fault, user should start the chip and do the initialization job.
- Command queue overrun interrupt
  - When chip report cmd_que_overrun_int interrupt, it means the chip command buffer is full. So the external system should stop sending command, and clear the interrupt
- Command queue valid interrupt
  - When chip report cmd_que_vld_int interrupt, it means the chip command buffer has not much command.
  - Usage: for the users, it should send next command, and clear the interrupt. The interrupt threshold can be changed by configured the cmd_que_int_trig register.
- result buffer overrun interrupt
  - When chip report result_buf_overrun_int interrupt, it means the nonce number exceeds the depth of memory, so the core will pause and report interrupt to user.

- User should fetch the result and make sure the number equals the depth of memory, after doing that, clear the interrupt

- result buffer valid interrupt
  - When the chip report this interrupt, it means the result buffer has sufficient nonce to return back to users.
  - User should fetch the result to avoiding the result_buf_overrun_int generating.

5.7 Reset control

- External reset
  - The Minion chip has one hardware reset signal, rstn_in, which is the global reset, this signal can reset whole chip.
- Software reset
  - In the addition, the MINION chip has many software reset signal to reset separate module, user can configure corresponding signal to reset module.
  - Sys_sw_rstn: reset all the module of chip.
  - Spi_sw_rstn: reset SPI interface module.
  - Cmd_que_sw_rstn: reset command buffer module.
  - Result_que_sw_rstn: reset result buffer module.
  - All_cores_sw_rstn: reset all cores module.
  - Sham_sw_rstn: reset SHA manager module.

5.8 LED control

- The MINION chip has 5 LED signal, and user can observe the status of chip easily.
- LED_SPI
  - This led is to test chip if it is good.
- LED_BUSY
  - This led is on when cores are working.
- LED_IDLE
  - This led is on when cores are not working.
- LED_NONCE
  - This led is on when cores get correct nonce.
- LED_HOT
  - This led is on when chip temperature is higher than threshold.

6. ELECTIRICAL SPECIFICATION and TIMING

6.1 DC Characteristics

Symbol	Description	Rating			Units
		MIN	TYP	MAX	
T_{JUN}	Junction temperature	-40		125	℃
T_{AMB}	Ambient temperature	-40		50	℃
VCC	Core voltage		0.85		V
V_{IO}	IO voltage		3.3		V
AVDD	Analog voltage		1.8		V
ESD: HBM	Human body model	2			KV
ESD: MM	Machine model	200			V

6.2 AC Timing Characteristics

Figure 6. SPI Timing