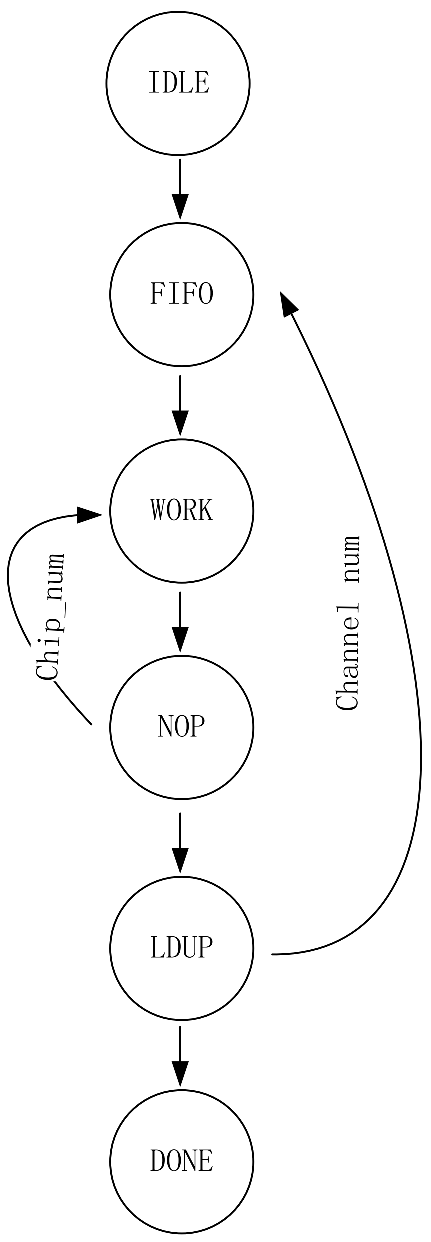
MM Update for A3222Q56

New Features:

- SPI-A3222 Interface;
- I2C to SHA256 Special Channel;
- I2C Data Interface;
- Support Firmware Reload Protocol;
- DNA Read Interface;

1. aLink

1.1 FSM



NOTE.

(1) Chip_num: the number of chip under a SPI String.

(2) Ch_num: the number of SPI String.

(3) Word_num: chip_num * 23;

1.2 Register Description (Base Address 0x80000500)

Register Name	Address	Address Offset within Register Word	Description
TXFIFO	0x00	TxFIFO Data Input, **MSB**, [W] A Message Block Include: Nonce[1w]+WORK[17w]+nonce2[2W]+cpm2[1w]+cpm1[1w]+cpm0[1w]	TxFIFO Data Input
RXFIFO	0x04	RxFIFO Data Output,[R] A Receive Block Include: Nonce2[2w]+nonce[1w]*8+{32'hbeaf_12xx}[1w], 8'hxx means channel number: 0 to ch_num-1;	RxFIFO Data Output

STATE	0x08	[0] TxFIFO Full;[R]
		[1] Flush;[W1]
		[11:2] TxFIFO Counter MAX 512+;(Word)[R]
		[12] Reserved;
		[15:13] State of FSM;[R]
		000: IDLE;
		001: WORK;
		010: NOP;
		011: DONE;
		[16] RxFIFO Empty;[R]
		[19:17] Reserved;
		[29:20] RxFIFO Counter MAX 512+;(Word)[R]
		[31:30] Reserved;
			State Register

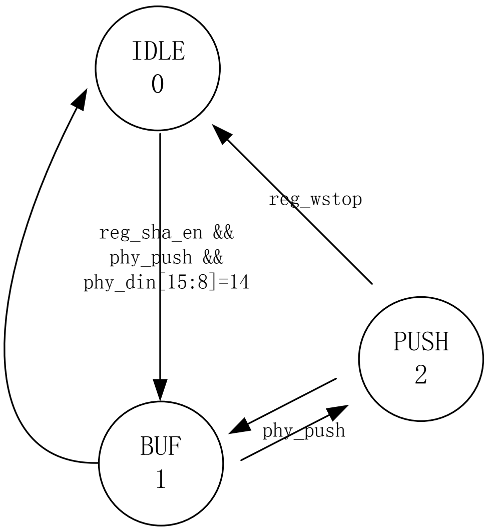
TIMEOUT	0x0c	Time Out Counter [WR]
		[27:0] Time Out Counter;
		100MHz: 0x4318c63;
		200MHz: 0x4318c63 / 2;
		
		[31:28] Reserved;
			Time Out Counter

SCK	0x10	[7:0] Half SCK Timing Register; Duty Cycle Always be 50%;
		[15:8] Reserved;
		[21:16] Channel Number;
		[23:22] Reserved;
		[31:24] Word Number: Chip Number * 23;
		Example: chip number = 4, then word number = 4 * 23 = 0x8c;
			SCK Register

2. ATWI [NEW Feature]

2.1 Register Description (Base Address 0x80000700)

Register Name	Address	Address Offset within Register Word	Description
CTRL	0x00	[8:0] rx FIFO count;[R] [17:9] tx FIFO count;[R] [18] ATWI Write Stop, Write 1 to clear;[RW] When this bit = 1, data in RxFIFO is valid for read. [19] ATWI Read Stop, Write 1 to clear;[RW] When this bit = 1, data in TxFIFO already sent to Mater. [20] ATWI Read Error, Write 1 to clear;[RW] When this bit = 1, a data read error occur, may set TxFIFO Reset. [21] Rx FIFO Reset ONLY;[W1] [22] Tx FIFO Reset ONLY;[W1] [23] Logic Reset ONLY;[W1] [24] RX Interrupt MASK Set, default 1;[W1R] [25] RX Interrupt MASK Clear;[W1] [27:26] Reserved; [30:28] state;[R] [31] Reserved; INT NOTE: (1) I2C Interrupt Number is [2] (2) When RX-COUNT > 0 a Interrupt will occur. (3) When Interrupt occur and MASK = 0, software may: set MASK=1, then read RX to zero until finish the int_function. [26] Enable Reboot[W1] [27] Enable Hardware Package Filler[WR]	

		If this bit enable, the MM's protocol CMD = 8'd14 payload(256bit) will switch to SHA256-POOL.  IDLE 0 reg_wstop reg_sha_en && phy_push && phy_din[15:8]=14 PUSH 2 BUF 1 phy_push	
ADDR	0x04	[6:0] Slave Address[RW]	
TX	0x08	[31:0] Tx FIFO [W]	
RX	0x0c	[31:0] Rx FIFO [R]	

3. DNA [NEW Feature]

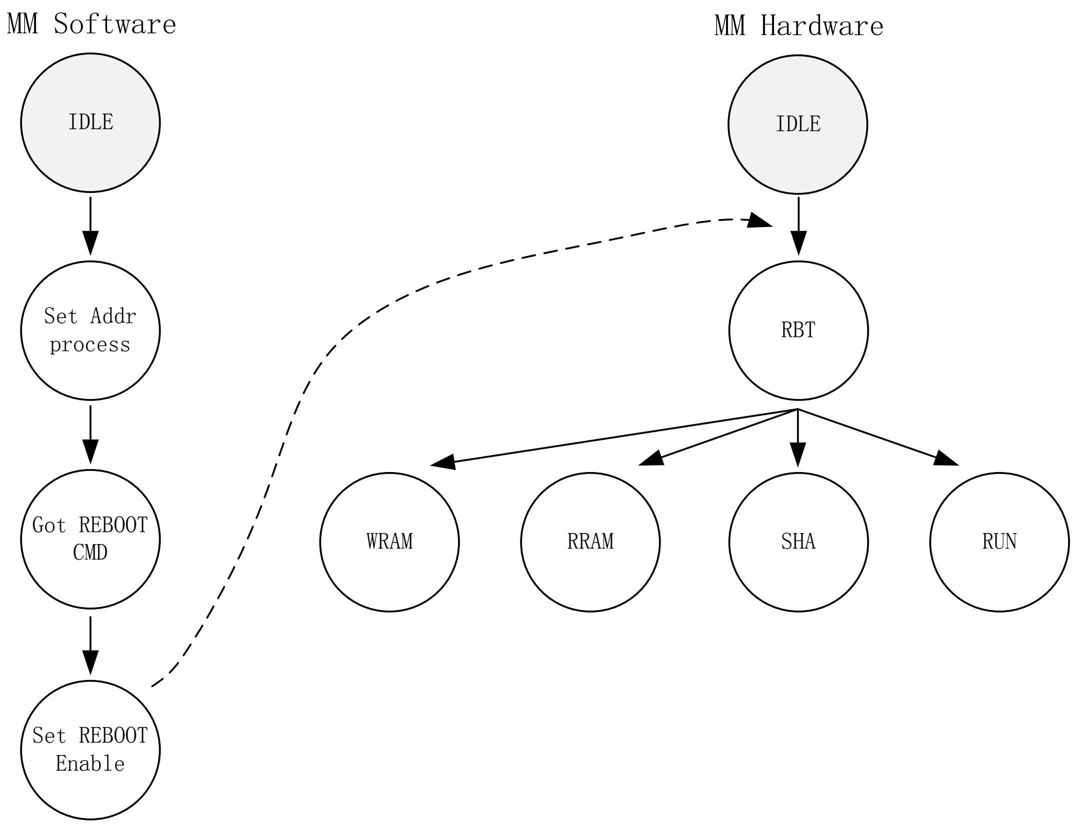
3.1 Register Description (Base Address 0x80000710)

Register Name	Address	Address Offset within Register Word	Description
DNA	0x0	[0] DNA Clock;[W] [1] DNA Data Input;[W] [2] DNA Read;[W] [3] DNA Shift;[W] [4] DNA Data Output; [R]	

```
void dna_rd(unsigned int *data){
		unsigned int i, tmp;
		data[0] = 0;
		data[1] = 0;

		writel(0, 0x80000710); //idle
```

```
        writel(0|4, 0x80000710);
        writel(1|4, 0x80000710);
        tmp = readl(0x80000710);
        data[1] = (data[1]<<1) | ((tmp >> 4)&1);
        writel(0|4, 0x80000710);//shift

        writel(0, 0x80000710); //idle
        for(i = 1; i < 32; i++){
                writel(0|8, 0x80000710);
                writel(1|8, 0x80000710);
                tmp = readl(0x80000710);
                data[1] = (data[1]<<1) | ((tmp >> 4)&1);
                writel(0|8, 0x80000710);
        }
        writel(0, 0x80000710); //idle

        for(; i < 57; i++){
                writel(0|8, 0x80000710);
                writel(1|8, 0x80000710);
                data[0] = (data[1]>>31) | (data[0] << 1);
                data[1] = (data[1]<< 1) | ((readl(0x80000710) >> 4)&1);
                writel(0|8, 0x80000710);
        }
}
```

4. REBOOT [NOT ENABLED BY SOFTWARE]



Command:
0x00000000: Write RAM:
{addr[15:0], 8’h00, 8’h00}: Write IRAM, the start iram address is addr[15:0];
{addr[15:0], 8’h01, 8’h00}: Write DRAM, the start dram address is addr[15:0];

0x00000001: Read RAM;
{addr[15:0], 8’h00, 8’h01}: Read IRAM, the start iram address is addr[15:0];
{addr[15:0], 8’h01, 8’h01}: Read DRAM, the start dram address is addr[15:0];

0x00000002: Read SHA256;
{16’h0, 8’h0, 8’h02}: Read SHA256 8words;
{16’h0, 8’h1, 8’h02}: Set sha256 initial;

0x00000003: RUN User Code;

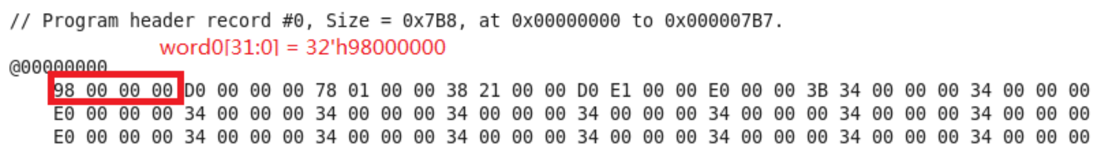
Gen mem file:
data2mem -bd ../mm.elf -d -o m mm.mem

$vi mm.mem //open memory file

IRAM:

IRAM: 4096KWord,

Address Zone: 12’h000, 12’h001, 12’h002 … 12’hfff

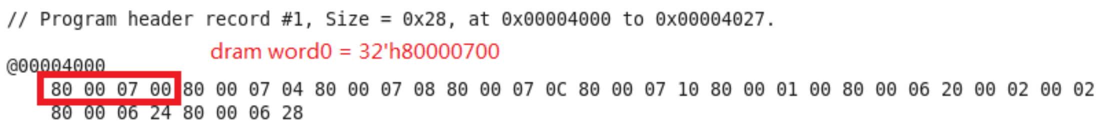


```
// Program header record #0, Size = 0x7B8, at 0x00000000 to 0x000007B7.
                word0[31:0] = 32'h98000000
@00000000
   98 00 00 00 D0 00 00 00 78 01 00 00 38 21 00 00 D0 E1 00 00 E0 00 00 3B 34 00 00 00 34 00 00 00
   E0 00 00 00 34 00 00 00 34 00 00 00 34 00 00 00 34 00 00 00 34 00 00 00 34 00 00 00 34 00 00 00
   E0 00 00 00 34 00 00 00 34 00 00 00 34 00 00 00 34 00 00 00 34 00 00 00 34 00 00 00 34 00 00 00
```

DRAM:

DRAM: 4096KWord,

Address Zone: 12’h000, 12’h001, 12’h002 … 12’hfff



```
// Program header record #1, Size = 0x28, at 0x00004000 to 0x00004027.
                   dram word0 = 32'h80000700
@00004000
   80 00 07 00 80 00 07 04 80 00 07 08 80 00 07 0C 80 00 07 10 80 00 01 00 80 00 06 20 00 02 00 02
   80 00 06 24 80 00 06 28
```

Example: Led twinkle firmware download:

(1) Enable RBOOT in the old firmware by: write I2C.CTRL[26] Enable Reboot[W1]

(2) New_fimeware define

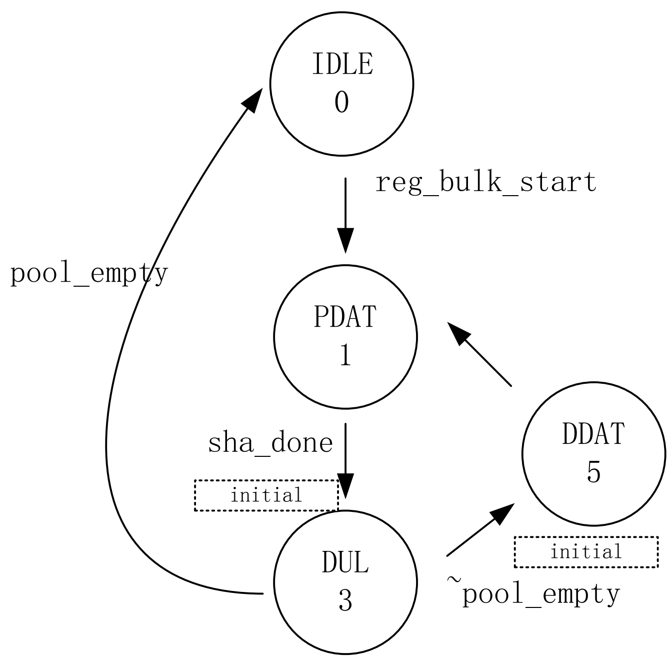
```
#define I2C_WI_LEN          (0x524+4)
#define I2C_WD_LEN          (0x14+4)

char txdat_iram[I2C_WI_LEN] = {0x00, 0x00, 0x00, 0x00,/*cmd: write iram*/
            0x98, 0x00, 0x00, 0x00, 0xD0, 0x00, 0x00, 0x00, 0x78, 0x01, 0x00,
0x00, 0x38, 0x21, 0x00, 0x00, 0xD0, 0xE1, 0x00, 0x00, 0xE0, 0x00, 0x00, 0x3B,
0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00,
            0xE0, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00,
0x00, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00,
0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00,
            0xE0, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00,
0x00, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00,
0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00,
            0xE0, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00,
0x00, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00,
0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00,
            0xE0, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00,
0x00, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00,
0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00,
            0xE0, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00,
0x00, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00,
0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00,
            0x5B, 0x9D, 0x00, 0x00, 0xF8, 0x00, 0x00, 0x1F, 0xF8, 0x00, 0x00,
0xF2, 0xE0, 0x00, 0x00, 0x2D, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00,
0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00,
            0xE0, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00,
0x00, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00,
0x34, 0x00, 0x00, 0x00, 0x34, 0x00, 0x00, 0x00,
            0x78, 0x1C, 0x00, 0x00, 0x3B, 0x9C, 0x7F, 0xFC, 0x78, 0x1A, 0x00,
0x00, 0x3B, 0x5A, 0xC0, 0x10, 0x78, 0x01, 0x00, 0x00, 0x38, 0x21, 0x40, 0x14,
0x78, 0x03, 0x00, 0x00, 0x38, 0x63, 0x44, 0x28,
            0x44, 0x23, 0x00, 0x04, 0x58, 0x20, 0x00, 0x00, 0x34, 0x21, 0x00,
0x04, 0xE3, 0xFF, 0xFF, 0xFD, 0x34, 0x01, 0x00, 0x00, 0x34, 0x02, 0x00, 0x00,
0x34, 0x03, 0x00, 0x00, 0xF8, 0x00, 0x00, 0x82,
            0x37, 0x9C, 0xFF, 0xC8, 0x5B, 0x81, 0x00, 0x04, 0x5B, 0x82, 0x00,
0x08, 0x5B, 0x83, 0x00, 0x0C, 0x5B, 0x84, 0x00, 0x10, 0x5B, 0x85, 0x00, 0x14,
0x5B, 0x86, 0x00, 0x18, 0x5B, 0x87, 0x00, 0x1C,
            0x5B, 0x88, 0x00, 0x20, 0x5B, 0x89, 0x00, 0x24, 0x5B, 0x8A, 0x00,
0x28, 0x5B, 0x9E, 0x00, 0x30, 0x5B, 0x9F, 0x00, 0x34, 0x2B, 0x81, 0x00, 0x38,
0x5B, 0x81, 0x00, 0x2C, 0xC3, 0xA0, 0x00, 0x00,
            0x2B, 0x81, 0x00, 0x04, 0x2B, 0x82, 0x00, 0x08, 0x2B, 0x83, 0x00,
0x0C, 0x2B, 0x84, 0x00, 0x10, 0x2B, 0x85, 0x00, 0x14, 0x2B, 0x86, 0x00, 0x18,
0x2B, 0x87, 0x00, 0x1C, 0x2B, 0x88, 0x00, 0x20,
            0x2B, 0x89, 0x00, 0x24, 0x2B, 0x8A, 0x00, 0x28, 0x2B, 0x9D, 0x00,
0x2C, 0x2B, 0x9E, 0x00, 0x30, 0x2B, 0x9F, 0x00, 0x34, 0x37, 0x9C, 0x00, 0x38,
0xC3, 0xC0, 0x00, 0x00, 0x37, 0x9C, 0xFF, 0xEC,
            0x5B, 0x8B, 0x00, 0x14, 0x5B, 0x8C, 0x00, 0x10, 0x5B, 0x8D, 0x00,
0x0C, 0x5B, 0x8E, 0x00, 0x08, 0x5B, 0x9D, 0x00, 0x04, 0x78, 0x0E, 0x00, 0x00,
0x39, 0xCE, 0x40, 0x14, 0x41, 0xC1, 0x00, 0x00,
            0x5C, 0x20, 0x00, 0x18, 0x78, 0x0D, 0x00, 0x00, 0x78, 0x0C, 0x00,
0x00, 0x39, 0xAD, 0x05, 0x1C, 0x39, 0x8C, 0x05, 0x18, 0xC9, 0xAC, 0x68, 0x00,
0x78, 0x0B, 0x00, 0x00, 0x15, 0xAD, 0x00, 0x01,
            0x39, 0x6B, 0x40, 0x18, 0x15, 0xAD, 0x00, 0x01, 0x29, 0x62, 0x00,
0x00, 0x35, 0xAD, 0xFF, 0xFF, 0x50, 0x4D, 0x00, 0x0A, 0x34, 0x42, 0x00, 0x01,
0xB4, 0x42, 0x08, 0x00, 0xB4, 0x21, 0x08, 0x00,
            0xB5, 0x81, 0x08, 0x00, 0x28, 0x21, 0x00, 0x00, 0x59, 0x62, 0x00,
0x00, 0xD8, 0x20, 0x00, 0x00, 0x29, 0x62, 0x00, 0x00, 0x55, 0xA2, 0xFF, 0xF8,
0x34, 0x01, 0x00, 0x01, 0x31, 0xC1, 0x00, 0x00,
            0x2B, 0x9D, 0x00, 0x04, 0x2B, 0x8B, 0x00, 0x14, 0x2B, 0x8C, 0x00,
0x10, 0x2B, 0x8D, 0x00, 0x0C, 0x2B, 0x8E, 0x00, 0x08, 0x37, 0x9C, 0x00, 0x14,
0xC3, 0xA0, 0x00, 0x00, 0x37, 0x9C, 0xFF, 0xFC,
            0x5B, 0x9D, 0x00, 0x04, 0x2B, 0x9D, 0x00, 0x04, 0x37, 0x9C, 0x00,
0x04, 0xC3, 0xA0, 0x00, 0x00, 0x37, 0x9C, 0xFF, 0xFC, 0x5B, 0x9D, 0x00, 0x04,
0x78, 0x01, 0x00, 0x00, 0x38, 0x21, 0x05, 0x20,
            0x28, 0x22, 0x00, 0x00, 0x44, 0x40, 0x00, 0x05, 0x78, 0x02, 0x00,
0x00, 0x38, 0x42, 0x00, 0x00, 0x44, 0x40, 0x00, 0x02, 0xD8, 0x40, 0x00, 0x00,
0x2B, 0x9D, 0x00, 0x04, 0x37, 0x9C, 0x00, 0x04,
            0xC3, 0xA0, 0x00, 0x00, 0x37, 0x9C, 0xFF, 0xFC, 0x5B, 0x9D, 0x00,
0x04, 0x2B, 0x9D, 0x00, 0x04, 0x37, 0x9C, 0x00, 0x04, 0xC3, 0xA0, 0x00, 0x00,
0x37, 0x9C, 0xFF, 0xF4, 0x5B, 0x8B, 0x00, 0x0C,
            0x5B, 0x8C, 0x00, 0x08, 0x5B, 0x9D, 0x00, 0x04, 0x78, 0x01, 0x00,
0x00, 0x38, 0x21, 0x05, 0x14, 0x28, 0x21, 0xFF, 0xFC, 0x34, 0x02, 0xFF, 0xFF,
0x44, 0x22, 0x00, 0x08, 0x78, 0x0B, 0x00, 0x00,
            0x39, 0x6B, 0x05, 0x10, 0x34, 0x0C, 0xFF, 0xFF, 0x35, 0x6B, 0xFF,
0xFC, 0xD8, 0x20, 0x00, 0x00, 0x29, 0x61, 0x00, 0x00, 0x5C, 0x2C, 0xFF, 0xFD,
0x2B, 0x9D, 0x00, 0x04, 0x2B, 0x8B, 0x00, 0x0C,
            0x2B, 0x8C, 0x00, 0x08, 0x37, 0x9C, 0x00, 0x0C, 0xC3, 0xA0, 0x00,
0x00, 0x37, 0x9C, 0xFF, 0xFC, 0x5B, 0x9D, 0x00, 0x04, 0x2B, 0x9D, 0x00, 0x04,
0x37, 0x9C, 0x00, 0x04, 0xC3, 0xA0, 0x00, 0x00,
            0xE0, 0x00, 0x00, 0x04, 0x34, 0x00, 0x00, 0x00, 0x34, 0x42, 0xFF,
0xFF, 0x5C, 0x40, 0xFF, 0xFE, 0x44, 0x20, 0x00, 0x04, 0x34, 0x21, 0xFF, 0xFF,
0x34, 0x02, 0x4E, 0x20, 0xE3, 0xFF, 0xFF, 0xFA,
            0xC3, 0xA0, 0x00, 0x00, 0x37, 0x9C, 0xFF, 0xFC, 0x5B, 0x9D, 0x00,
0x04, 0x34, 0x01, 0x03, 0xE8, 0xFB, 0xFF, 0xFF, 0xF4, 0x34, 0x01, 0x00, 0x02,
0xF8, 0x00, 0x00, 0x44, 0x34, 0x01, 0x03, 0xE8,
            0xFB, 0xFF, 0xFF, 0xF0, 0x34, 0x01, 0x00, 0x01, 0xF8, 0x00, 0x00,
0x40, 0xE3, 0xFF, 0xFF, 0xF8, 0x78, 0x01, 0x00, 0x00, 0x38, 0x21, 0x40, 0x00,
0x28, 0x23, 0x00, 0x00, 0x78, 0x02, 0x00, 0x00,
            0x78, 0x01, 0x00, 0x00, 0x38, 0x21, 0x40, 0x1C, 0x38, 0x42, 0x40,
0x24, 0xE0, 0x00, 0x00, 0x09, 0x28, 0x24, 0x00, 0x00, 0x40, 0x65, 0x00, 0x00,
0xB4, 0x44, 0x20, 0x00, 0x30, 0x85, 0x00, 0x00,
            0x28, 0x24, 0x00, 0x00, 0x34, 0x84, 0x00, 0x01, 0x20, 0x84, 0x03,
0xFF, 0x58, 0x24, 0x00, 0x00, 0x40, 0x64, 0x00, 0x05, 0x20, 0x84, 0x00, 0x01,
0x5C, 0x80, 0xFF, 0xF6, 0x34, 0x01, 0x00, 0x08,
            0xD0, 0x41, 0x00, 0x00, 0xC3, 0xA0, 0x00, 0x00, 0x20, 0x21, 0x00,
0xFF, 0x5C, 0x20, 0x00, 0x08, 0x78, 0x02, 0x00, 0x00, 0x38, 0x42, 0x40, 0x04,
0x28, 0x41, 0x00, 0x00, 0x28, 0x21, 0x00, 0x00,
            0x20, 0x21, 0x00, 0x02, 0x38, 0x21, 0x01, 0x00, 0xE0, 0x00, 0x00,
0x09, 0x78, 0x03, 0x00, 0x00, 0x38, 0x63, 0x40, 0x04, 0x28, 0x61, 0x00, 0x00,
0x28, 0x22, 0x00, 0x00, 0x78, 0x01, 0x00, 0x02,
            0xA0, 0x41, 0x08, 0x00, 0x78, 0x02, 0x01, 0x00, 0xB8, 0x22, 0x08,
0x00, 0x78, 0x03, 0x00, 0x00, 0x38, 0x63, 0x40, 0x04, 0x28, 0x62, 0x00, 0x00,
0x58, 0x41, 0x00, 0x00, 0xC3, 0xA0, 0x00, 0x00,
            0x37, 0x9C, 0xFF, 0xFC, 0x5B, 0x9D, 0x00, 0x04, 0x34, 0x01, 0x00,
0x00, 0xFB, 0xFF, 0xFF, 0xE7, 0x34, 0x01, 0x00, 0x20, 0xD0, 0x41, 0x00, 0x00,
0x2B, 0x9D, 0x00, 0x04, 0x37, 0x9C, 0x00, 0x04,
            0xC3, 0xA0, 0x00, 0x00, 0x37, 0x9C, 0xFF, 0xFC, 0x5B, 0x9D, 0x00,
0x04, 0x34, 0x01, 0x00, 0x01, 0xFB, 0xFF, 0xFF, 0xDE, 0x34, 0x01, 0x00, 0x20,
0xD0, 0x41, 0x00, 0x00, 0x2B, 0x9D, 0x00, 0x04,
            0x37, 0x9C, 0x00, 0x04, 0xC3, 0xA0, 0x00, 0x00, 0x78, 0x02, 0x00,
0x00, 0x78, 0x03, 0x00, 0x00, 0x20, 0x21, 0x00, 0xFF, 0x38, 0x42, 0x44, 0x24,
0x38, 0x63, 0x40, 0x0C, 0x30, 0x41, 0x00, 0x00,
            0x28, 0x62, 0x00, 0x00, 0x38, 0x21, 0x00, 0x02, 0x58, 0x41, 0x00,
0x00, 0xC3, 0xA0, 0x00, 0x00, 0x37, 0x9C, 0xFF, 0xF8, 0x5B, 0x8B, 0x00, 0x08,
0x5B, 0x9D, 0x00, 0x04, 0x90, 0x40, 0x08, 0x00,
            0x90, 0x20, 0x58, 0x00, 0xA1, 0x61, 0x58, 0x00, 0x21, 0x61, 0x00,
0x08, 0x44, 0x20, 0x00, 0x02, 0xFB, 0xFF, 0xFF, 0xB0, 0x21, 0x61, 0x00, 0x20,
0x44, 0x20, 0x00, 0x02, 0xFB, 0xFF, 0xFF, 0xD9,
            0x21, 0x6B, 0x00, 0x40, 0x45, 0x60, 0x00, 0x02, 0xFB, 0xFF, 0xFF,
0xDF, 0x2B, 0x9D, 0x00, 0x04, 0x2B, 0x8B, 0x00, 0x08, 0x37, 0x9C, 0x00, 0x08,
0xC3, 0xA0, 0x00, 0x00, 0x37, 0x9C, 0xFF, 0xFC,
            0x5B, 0x9D, 0x00, 0x04, 0xFB, 0xFF, 0xFF, 0x63, 0xFB, 0xFF, 0xFF,
0x74, 0x2B, 0x9D, 0x00, 0x04, 0x37, 0x9C, 0x00, 0x04, 0xC3, 0xA0, 0x00, 0x00,
0x37, 0x9C, 0xFF, 0xFC, 0x5B, 0x9D, 0x00, 0x04,
            0xFB, 0xFF, 0xFF, 0x2F, 0x2B, 0x9D, 0x00, 0x04, 0x37, 0x9C, 0x00,
0x04, 0xC3, 0xA0, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00,
0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00};
    char txdat_dram[I2C_WD_LEN] = {0x00, 0x00, 0x01, 0x00, /*cmd: write dram*/
            0x80, 0x00, 0x01, 0x00, 0x80, 0x00, 0x06, 0x20, 0x00, 0x02, 0x00,
0x02, 0x80, 0x00, 0x06, 0x24, 0x80, 0x00, 0x06, 0x28};

    char txdat_dram[4] = {0x00, 0x00, 0x00, 0x03/*cmd: run*/};
```

(3) Write CMD_IRAM: write txdat_iram;
(4) Write CMD_DRAM: write txdat_dram;
(5) Write CMD_RUN: write 0x00000003;

5. Bulk Double-SHA256 [NEW Feature]

3.1 Register Description (Base Address 0x80000400)

Register Name	Address	Address Offset within Register Word	Description
CMD	0x0	[4] initial H0~7 with default value [W1] [5] BULK SHA256 Start.[W1] 1st. Write 256bit pre data to sha256. 2nd. Write BULK-SHA256-START. 3rd. Wait DONE. [14:6] POOL Data Depth, count by word.[R] The Max is 256words. [15] POOL Reset[W1] [16] Bulk Sha done, write 1 to clear[WR]	


IDLE
0
reg_bulk_start
pool_empty
PDAT
1
sha_done
initial
DDAT
5
DUL
3
initial
~pool_empty

6. SFTB (Shifter B)

6.1 Register Description (Base Address 0x80000600)

Register Name	Address	Address Offset within Register Word	Description
SFTB	0x2c	Shift Controller [1:0] Shifter Command:[W] 00: Set master reset.[Avalon4 NOT support] 01: Shift register. 10: Storage register. 11: Output enable. [2] Output enable value, Only valid when Shifter Command == 2'b11(Low Active),default=1. [3] Done. Write 1 to clear.[RW1] [7:4] Reserved. [15:8] Data Register. [31:16] Reserved.	Shift Controller B

7. SFTC (Shifter C)

7.1 Register Description (Base Address 0x80000600)

Register Name	Address	Address Offset within Register Word	Description

SFTC	0x30	[3:0] LED_RED
		0000: set 0;
		0001: set 1;
		0010: set 0 -> 1 -> 0;
		0011: set 0 -> 1 -> 0, while(1);
		0100: reserved.
		0101: breath
			Shift Controller C

		[7:4] LED_YELLOW
		0000: set 0;
		0001: set 1;
		0010: set 0 -> 1 -> 0;
		0011: set 0 -> 1 -> 0, while(1);	

		0100: reserved. 0101: breath	

		[11:8] LED_GREEN0
		0000: set 0;
		0001: set 1;
		0010: set 0 -> 1 -> 0;
		0011: set 0 -> 1 -> 0, while(1);
		0100: IIC TX Bling-Bling, By Hardware.
		0101: breath	

		[15:12] LED_GREEN1
		0000: set 0;
		0001: set 1;
		0010: set 0 -> 1 -> 0;
		0011: set 0 -> 1 -> 0, while(1);
		0100: IIC RX Bling-Bling, By Hardware.
		0101: breath	

		[19:16] LED_GREEN2
		0000: set 0;
		0001: set 1;
		0010: set 0 -> 1 -> 0;
		0011: set 0 -> 1 -> 0, while(1);
		0100: NONCE 0~4 Bling-Bling, By Hardware.
		0101: breath	

		[23:20] LED_GREEN3
		0000: set 0;
		0001: set 1;
		0010: set 0 -> 1 -> 0;
		0011: set 0 -> 1 -> 0, while(1);
		0100: NONCE 5~9 Bling-Bling, By Hardware.
		0101: breath	

		[3:0] LED_GREEN4
		0000: set 0;
		0001: set 1;
		0010: set 0 -> 1 -> 0;
		0011: set 0 -> 1 -> 0, while(1);
		0100: reserved.
		0101: breath	

		[3:0] LED_GREEN5
		0000: set 0;
		0001: set 1;
		0010: set 0 -> 1 -> 0;
		0011: set 0 -> 1 -> 0, while(1);
		0100: reserved.
		0101: breath	

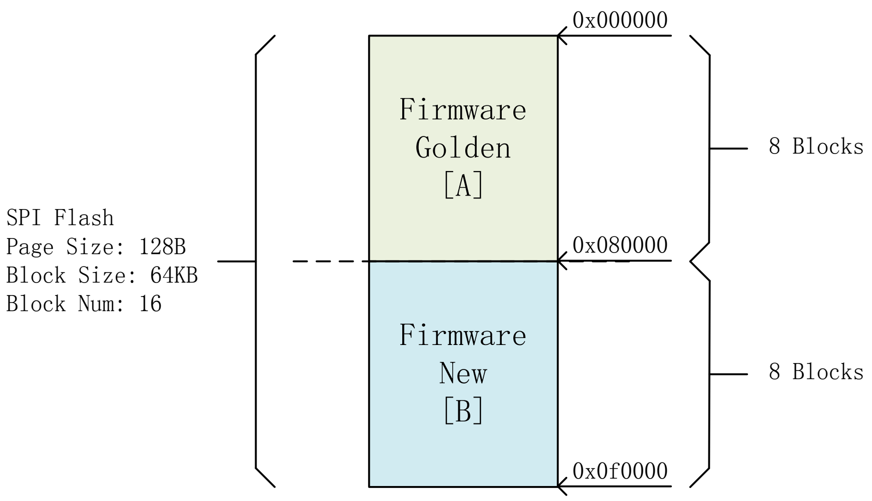
8. GPIO

8.1 Register Description (Base Address 0x80000600)

Register Name	Address	Address Offset within Register Word	Description

GPIO	0x24	[0] Output, Front Panel 74hc164 Shift Data;
		[1] Output, Front Panel 74hc164 Clock;
		[2] Output, Port A Reset, Low Active;
		[3] Output, Port B Reset, Low Active;
		[4] Output, LED1;
		[5] Output, LED2;
		[6] Output, LED3;
		[7] Output, LED4;
		[15:8] Reserved for GPIO Output;
		
		[16] Input, Port A Power Good 1.
		[17] Input, Port A Power Good 2.
		[18] Input, Port A Power Good 3.
		[19] Input, Port A Power Good 4.
		[20] Input, Port A Power Good 5.
		[21] Input, Port B Power Good 1.
		[22] Input, Port B Power Good 2.
		[23] Input, Port B Power Good 3.
		[24] Input, Port B Power Good 4.
		[25] Input, Port B Power Good 5.
		[31:26] Reserved for GPIO Input;
			Only For Avalon4

9. MultiBoot[NOT ENABLED BY SOFTWARE]



8.1 Register Description (Base Address 0x80000800)

Register Name	Address	Address Offset within Register Word	Description
FLASH	0x00	[5] MBOOT_MISO [R] [4] MBOOT_SCL, [3] MBOOT_CS, [2] MBOOT_MOSI, [1] MBOOT_HOLD_N, [0] MBOOT_WP_N	

Register Name	Address	Address Offset within Register Word	Description
ICAP_O	0x04	[18] ICAP_CLK, [17] ICAP_CE, [16] ICAP_WRITE, [15:0] ICAP_I[15:0]	

Register Name	Address	Address Offset within Register Word	Description
ICAP_I	0x08	[16] ICAP_BUSY, [R] [15:0] ICAP_O[15:0], [R]	

8.2 ICAP MultiBoot Example

```
int icap_enable(void){
		*icap_o = 0x00000;
		return 0;
}

int icap_disenable(void){
		*icap_o = 0x70000;
		return 0;
}

int icap_write_16bit(unsigned int data_o){
		*icap_o = 0x00000 | (data_o & 0xffff);
		*icap_o = 0x40000 | (data_o & 0xffff);
		*icap_o = 0x00000 | (data_o & 0xffff);
		return 0;
}

int icap_mboot_start(
unsigned int mboot_file0_addr_start,
unsigned int mboot_file1_addr_start
){
		delay(1000);
		icap_enable();
		icap_write_16bit(0x2000);
		icap_write_16bit(0x2000);
		icap_write_16bit(0x2000);
		icap_write_16bit(0x2000);

		icap_write_16bit(0xffff);
		icap_write_16bit(0xaa99);
		icap_write_16bit(0x5566);
		icap_write_16bit(0x31e1);
		icap_write_16bit(0xffff);
		icap_write_16bit(0x3261);
		icap_write_16bit(mboot_file1_addr_start & 0xffff);//MultiBoot Start Address[15:0]
		icap_write_16bit(0x3281);
		icap_write_16bit(((mboot_file1_addr_start >> 16) & 0xff) | 0x0300);//Opcode and
MultiBoot Start Address[23:16]
		icap_write_16bit(0x32a1);
		icap_write_16bit(mboot_file0_addr_start & 0xffff);//FallBack Start Address [15:0]
		icap_write_16bit(0x32c1);
```

```
        icap_write_16bit(((mboot_file0_addr_start >> 16) & 0xff) | 0x0300);//Opcode and
Fallback Start Address [23:16]

        icap_write_16bit(0x32e1);
        icap_write_16bit(0x0000);

        icap_write_16bit(0x30a1);
        icap_write_16bit(0x0000);

        icap_write_16bit(0x3301);
        icap_write_16bit(0x2100);

        icap_write_16bit(0x3201);
        icap_write_16bit(0x001f);

        icap_write_16bit(0x30a1);
        icap_write_16bit(0x000e);

        icap_write_16bit(0x2000);
        icap_write_16bit(0x2000);
        icap_write_16bit(0x2000);
        icap_write_16bit(0x2000);

        icap_disable();
        delay(5000);
        return 0;
}
```