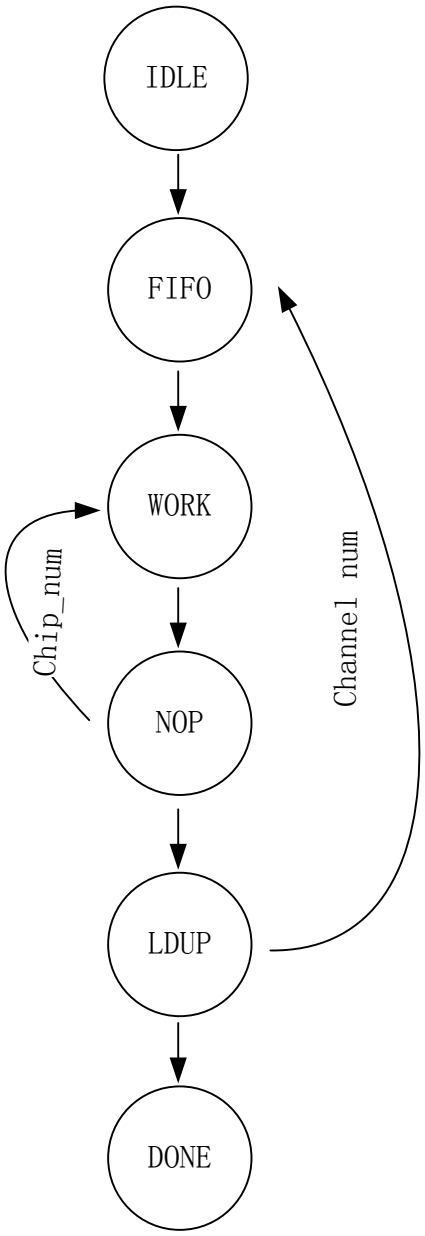
MM Update for A3222Q56

1. aLink[Update MM DOC Chapter 5]

1.1 FSM



NOTE.

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

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]+nocne[1w]	RxFIFO Data Output
STATE	0x08	[0] TxFIFO Full; [1] Flush; [11:2] TxFIFO Counter MAX 1024;(Word) [12] Reserved; [15:13] State of FSM; [16] RxFIFO Empty; [19:17] Reserved; [28:20] RxFIFO Counter MAX 512;(Word) [31:29] Reserved;	State Register
TIMEOUT	0x0c	Time Out Counter [WR] [24:0] Time Out Counter; [31:25] 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; [29:24] Chip Number; [31:30] Reserved	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] [30:24] Slave Address [R]	
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);
	}
}
```