AM BE200 Datasheet

The AM BE200 is a Bitcoin hashing chip with the design target of high performance per watt.

Specifications:

- 40nm TSMC process.
- Standard SPI interface.
- 32 highly optimized hashing cores in single chip.
- The typical hash power is ~12GH/s in rated mode.

Pins:

RESET	Reset pin
OSC_CLK	Input clock of PLL (20MHz Typical, 1.8v)
BS	BS for PLL configuration(see PLL document for details), 0 is ground, 1 is 1.8v.
{HARD1, HARD0}	Two pins for configuration hard as follows: hard[1:0]=2'b00, diff=1 hard[1:0]=2'b01, diff=64 hard[1:0]=2'b10, diff=4096 hard[1:0]=2'b11, diff=262144 0 is ground, 1 is 1.8v.
SS	SS pin for SPI, low active
SCK	SCK pin, spi clock provided by the master (usually MCU), with the frequency range of 0-20MHz
MOSI	MOSI pin for spi(input)
MISO	MISO pin for spi(output)
VDD_CORE	Core Vdd, 0.55v~0.88v
PLLVSS	Common ground of PLLPOC and PLLDVDDCORE
PLLVDDCORE	PLL Core Vdd, 0.9v
PLLDVSS	PLL Levelshifter Ground
PLLDVDD	PLL Levelshifter Vdd, 0.9v
PLLAHVSS	PLL Analog Ground
PLLAHVDD	PLL Analog Vdd, 1.8v
PLLPOC	PLL Power Control Vdd, 1.8v
VDD_IO	IO Vdd, 1.8v
VDD_IO(POC)	IO Power Control Vdd, 1.8v

Since the chip package is QFN64, all VSS except for PLL are downbonded to the center pad. PLLVSS can be connected directly to the center pad. PLLDVDD and PLLVDDCORE can be connected directly to each other. PLLPOC and (VDD_IO/VDD_IO(POC)) can be connected to each other. PLLAHVDD and (VDD_IO/ VDD_IO(POC)) are NOT recommended to be connected directly to each other. PLLAHVSS and other grounds are also NOT recommended to be connected directly to each other.

Protocols:

The chip employs a standard SPI interface to configure the chip. The SPI is only working in mode 0(CPOL=0, and CPHA=0, sampling in posedge, modifying the register in negedge). There are two types of SPI transmissions: CMD transmission and DATA transmission. Both types are of 1 byte(8 bits). Each DATA transmission must follow a CMD transmission. However, not all CMD transmissions are followed by DATA transmissions. Only RD(read) and WR(write) commands are followed by DATA transmissions. The 2 MSB bits of CMD transmission encode the command type. The 6 LSB bits are the address（for RD/WR commands）. See the following table for details.

CMD type	2 MSB bits	6 LSB bits
CMD_CK(check status)	2'b00	Any values
CMD_RD(read regs)	2'b01	read reg address
CMD_WR(write regs)	2'b10	write reg address
CMD_RST(softreset chip)	2'b11	Any values

The chips MISO pin transports the MSB bit first. If we define the transported 8 bits as data[7:0], then the chip MISO pin first to transport the data[7], then data[6], and data[0] at last.
The CMD_CK checks the status register, and gets the result in the same transmission from the MISO pin, the status bit are transported as follows:

w_allow=data[0], r_ready=data[1], nonce_mask=data[5:2]

If w_allow is 1, the chip is ready for a new task, and the microcontroller can write the task via the WR command.
If r_ready is 1, the chip has a new nonce, and the microcontroller can read the nonce via the RD command.
To reduce the polling frequency, the chip has a buffer for 4 nonces. nonce_mask indicates which nonce register group has a nonce(if it is 1, the corresponding group has a nonce). Please check the address for details.

Registers:

All registers of each address are 8 bits. The register file (a total of 64 registers) itself is small endian.

Reg Address	Usage
0~43	Task address. 0~31 are for midstate(small endian), 32~43 are for data(small endian).
44	Dummy register. Writing to this address with any value starts hashing cores. w_allow will turn to low until the task finishes.

45	PLL configure register. Name this reg as pll_conf, then the PLL parameter will be:
	F6~F0=pll_conf[6:0];//default(after reset):7'b0100111;
	PD=pll_conf[7];//default 1'b0
	please check the PLL document for more details.

46~49	Nonce register group0(small endian) corresponding to nonce_mask[0]
50~53	Nonce register group1(small endian) corresponding to nonce_mask[1]
54~57	Nonce register group2(small endian) corresponding to nonce_mask[2]
58~61	Nonce register group3(small endian) corresponding to nonce_mask[3]
62	Dummy register. Reading from this address cleans r_ready and the nonce_mask. User should read this address after reading all nonces.
63	Status register with value of {2'b0, nonce_mask, r_ready, w_allow}. User can either use the CMD_CK or read this register to check status.

Note: The task addresses can be written in separate. If two tasks are only different in one register (for example, the task are generated by n-time rolling), after the former task finishes(w_allow being high), the user can just write the part of difference to reduce bandwidth consumption. But if you soft reset the chip, the task register will be cleaned so user should write to all task registers.

More Details:

The basic mechanism in chip is as follows:

1. The w_allow will turn to high when: 1)reset, 2)soft_reset, or 3)a task have been finished(Whole 2^32 nonce space has been traversed). The high value w_allow indicate the chip is ready for a new task. User can write task to the chip by writing to address 0~43.
2. After writing to address 44, The w_allow turns to low, and the chip starts calculating the nonce. When a nonce calculated, the r_ready will be high, which indicates at least one

nonce is ready for reading. User can read nonces based on the nonce mask information.
3. During the reading nonce, the chip still calculates the nonce when not all of the 2^32 space is traversed yet. If the traversing is finished, the w_allow will be high again, and the chip is ready for the next task.
4. During the calculation, a soft_reset will stop the current task, and the w_allow will be high for a new task.

Basic mining procedure for sofware:

1. Set the chip's HARD0, HARD1 and BS to demanded value. They could also be hard-coded in the hardware.
2. Set the chip's SS pin to low and other chips' SS pins to high.
3. Configure the PLL by writing to the address 45(for details of configuration, please check the PLL document).
4. Use CMD_CK or read the address 63 to check w_allow.
5. If w_allow is high, write the task to the chip(The addresses are 0~43), then write anything to the address 44 to start the chip.
6. Use CMD_CK or read the address 63 to check r_ready and nonce_mask information.
7. When r_ready is high, read the nonce based on nonce_mask.
8. Clear r_ready and nonce_mask by reading from address 62.
9. Use CMD_CK or read the address 63 to check if there is new nonce(r_ready) and if the task has been finished(w_allow).

PLL document:

bs decides the operating mode. When bs=0, the range of core clock frequency is 200MHz-400MHz. When bs=1, the range of core clock frequency is 375MHz-750MHz.

PD will turn to 0 whenever reset or soft_reset is triggered. The user should set PD to 1 (then to 0) to start the PLL again. When configuring the PLL without reset or soft_reset, the user should first set PD to 1 along with modifying other register bits, then set PD to 0 (while keeping other register bits). There is a time period of 0.2ms after PD turns to 0 and before PLL resumes working properly.

The core clock frequency is calculated as follows:
CLK_CORE=CLK_OSC*(F6:F0+1)/2.
For example, in default setting, F6:F0=0100111(39). If the frequency of the oscillator is 20MHz, then the frequency of the core is 20*(39+1)/2=400MHz.