INNOSILICON

A9 Zmaster
ASIC Datasheet Brief

Product Specification

V0.1

Notice

DISCLAIMER

INNOSILICON hereby provides the quality information to you but makes no claims, promises or guarantees about the accuracy, completeness, or adequacy of the information herein. The information contained herein is provided on an "AS IS" basis without any warranty, and INNOSILICON assumes no obligation to provide support of any kind or otherwise maintain the information. INNOSILICON disclaims any representation that the information does not infringe any intellectual property rights or proprietary rights of any third Parties. INNOSILICON makes no other warranty, whether express, implied or statutory as to any matter whatsoever, including but not limited to the accuracy or sufficiency of any information or the merchantability and fitness for a particular purpose. Neither INNOSILICON nor any of its representatives shall be liable for any cause of action incurred to connect to this service.

INNOSILICON TECHNOLOGY LTD

Skype: inno.miner

Email: sales@innosilicon.com and sales@innosilicon.com.cn

URL: www.innosilicon.com and www.innosilicon.com.cn

REVISION HISTORY

VERSION	DATE	DESCRIPTION
V0.1	Jun, 2018	Preliminary

CONTENT

1 GENERAL DESCRIPTION

The A9 Zmaster ASIC (also referred to as A9 Zmaster) is a simple to use, highly optimized GPU like ASIC hardware designed specifically for the purpose of the equihash with the highest power efficiency to-date. It is developed in the most advanced process nodes with very innovative architecture and low power consumption. A9 Zmaster is highly integrated GPU like solution using very few external components. It is extremely easy to use and easy to deploy in large scale with low cost.

With sophisticated on chip ADC and dynamic temperature and frequency scaling for highly reliable performance under all conditions, the A9 Zmaster has operation modes from low power to over clock high performance through adjustment of PLL output clock frequency and/or in combination with voltage adjustment. It is up to the miner manufacturers to decide what speed and power efficiency they design the miners for and/or whether it is configurable.

1.1 MAIN FEATURES

- Custom developed large scale GPU ASIC chip on advanced process using low power techniques for best efficiency with up to 1375 sol/s per chip with only 16W
- Custom flipchip IC package for lowest IR drop and high current feeding
- Configurable in SPI daisy chain for up to 128 ASICs under one MCU control
- Standard SPI interface with CS pin needed to enclose all useful SDI and SDO data.
- Flexibility to configure the PLL clock and computation performance.
- Built-in dynamic frequency/temperature scaling & high precision on chip monitoring ADC/sensor for reliability protection and automatic performance adjustment for maximized performance
- Built-in chip self tests and on the fly debug modes
- Regular top and bottom heat sinks plus fan cooling

1.2 HASHING RATE

PLL Output Clock Frequency (MHz)	Hash Rate (sol/s/Chip)	Power (W/Chip)	Core Voltage (V)
1250	1375	16.0	0.87
1200	1320	14.8	0.85
1100	1210	13.8	0.82

1.3 SPI DAISY CHAIN CONFIGURATION

The A9 Zmaster to/from MCU uses an industry standard SPI interface (except there is no tri-state and inter-A9 Zmaster SPI communication is proprietary) that enables a flexible daisy chaining up to 16 units. In addition to the support of the traditional DCDC driven PCB scheme (we called DCDC-supply parallel mode, shown in Figure 1-1).

The first chip in the daisy chain is connected to the upstream MCU controller using a standard SPI interface with a chip select CS pin. The 2nd chip downstream is connected to the 1st chip though the Innosilicon proprietary SPI interface (compatible with standard SPI port in CPU/MCU). Note that all SPI timing re-sync and re-alignment is done inside of each A9 Zmaster so if the input timing meets standard SPI timing requirement specified in the document, that A9 Zmaster's performance is guaranteed. At the end of the chain, the last chip should connect the SDO_R to SDI_R signals directly. Users are free to debug the loop anywhere in the middle of the chain by cutting off the rest of chain and debug the chips in a shorter loop. We recommend using two zero ohm resistors in between of each A9 Zmaster SPI's SDI and SDO so that the SPI chain can be configured into a shorter loop back chain after each A9 Zmaster so that debug can be done one A9 Zmaster at a time. We also added a special IO output for each A9 Zmaster to go high after the BIST command completion for PCB testing/indication purpose.

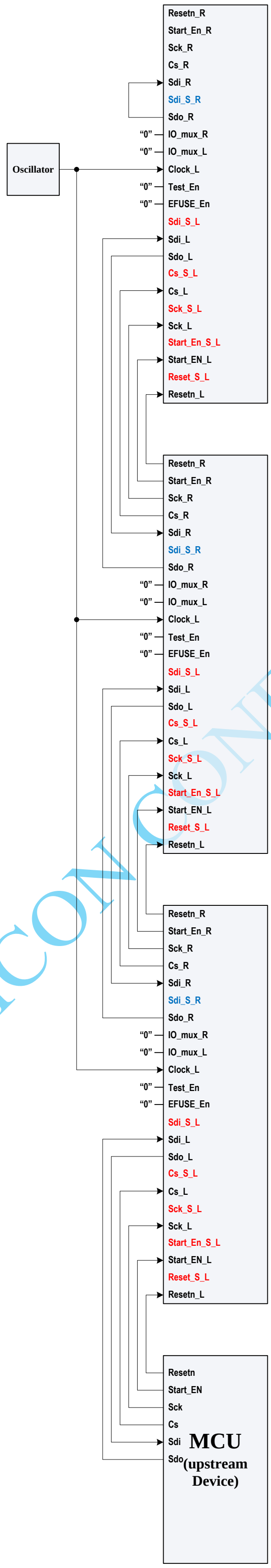

Figure 1-1 A9 Zmaster supports DCDC-supply **parallel mode** daisy chaining

Operating voltage: A9 Zmaster core voltage can be anywhere from 0.75V to 0.90V. IO voltage works at 1.8V. Due to high current (causing IR drop) nature in PCB which caused voltage loss, we suggest for the User PCB and packaging, the Vdd_core supply voltage not to be designed less than 0.75V. In the high performance hashing mode, our recommended operation range is centered around 0.90V (+-10%).

At the DCDC-supply parallel mode, the DCDC ripple for Vdd_core voltage should be less than 3% fluctuation, i.e., for 0.75V, the DCDC ripple should be less than 12mV.

Proper voltage/Level Shifters for all A9 Zmaster signal IO: Note that A9 Zmaster IOs are 1.8V based and all input signals need to be properly driven at 1.8V level. For all A9 Zmaster 1.8V IO such as SPI and reset, it is important that users use a proper level shifter with enough drive to handle 1.8V A9 Zmaster IO to/from 2.5V/3.3V domain, resistor level shifter won't work. Reset network can be tied together or separately controlled depending on customer choice. Please see our attached PCB reference design for parallel DCDC mode.

Clean Input Clock Reference for A9 Zmaster: For Clock input as reference to A9 Zmaster PLL, we recommend a point to point clean clock input. It is at user's risk to drive multi-input of A9 Zmaster using one clock source. We have no timing synchronization requirement for different A9 Zmasters on one PCB.

SPI Daisy Chain Interface: All A9 Zmaster SPI interface IOs are not tri-state-able so it is important that users use mux select outside of A9 Zmaster domain if they desire to drive multiple A9 Zmaster using a parallel SPI interface using CS pin. Remember to insert 0x0000s after each command to push the command response back to the MCU or Pi as SPI chain data needs to be looped back.

2 PACKAGE INFORMATION

2.1 CHIP PACKAGE

The A9 Zmaster Dominator package is a flipchip and shown in the Figure 2-1. The SPI interface on the upper is connected to the MCU or upstream device meanwhile the SPI interface on the below communicates with the downstream A9 Zmaster device. Apart from the data and control signals, other pads are core or IO supply voltage pads.

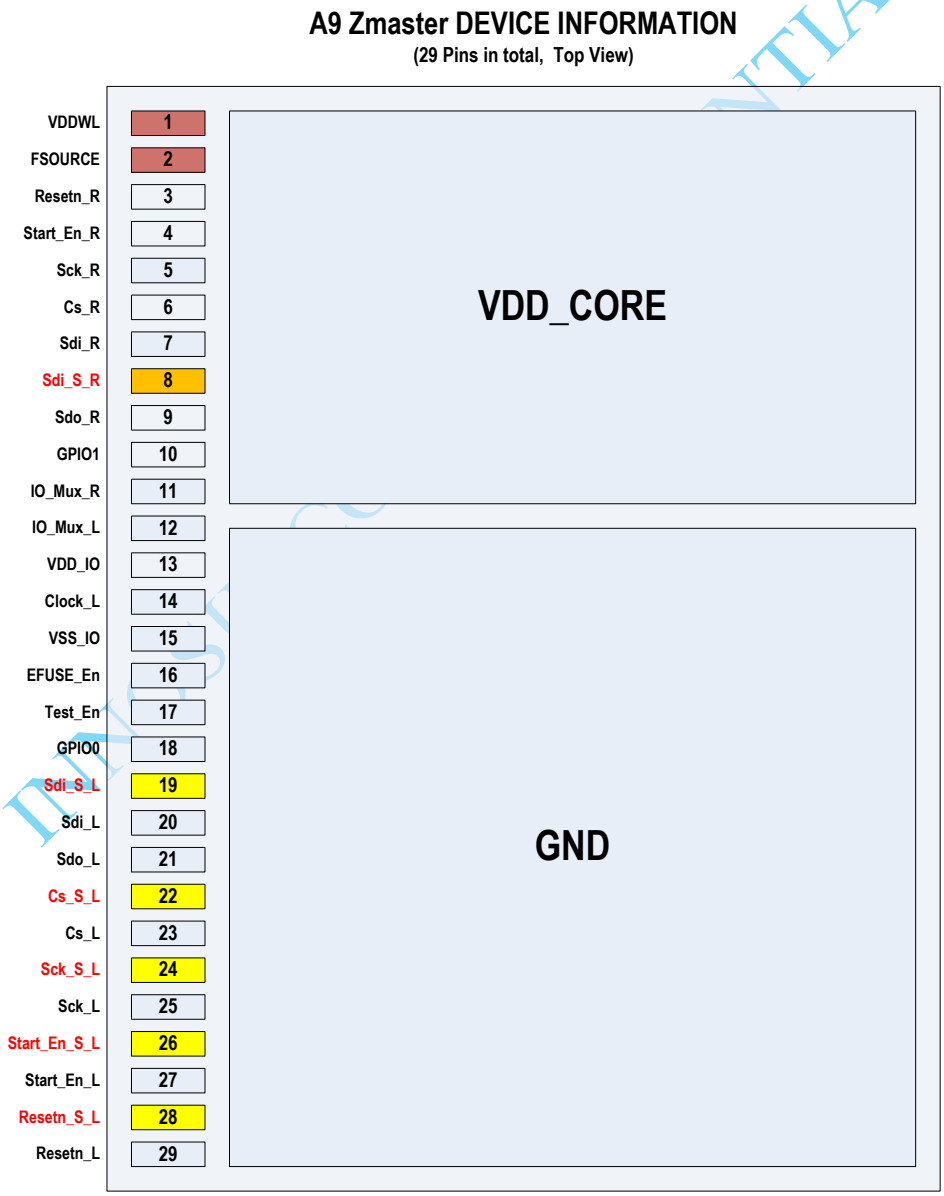

Figure 2-1 Chip package

2.2 PIN DESCRIPTION

The pin description is listed in the Table 2-1 below.

Table 2-1 Pin description

NAME	DIRECTION	PIN NUMBER	DESCRIPTION
Analog PINs			
VDD_IO	I/O	13	IO supply voltage at 1.8V
VSS_IO	I/O	15	IO Ground
VDD_WL	I/O	1	"High" to blow fuses, "Low" at all other time
FSOURCE	I/O	2	"High" to blow fuses, "Low" at all other time
Digital PINs			
Sck_L	I	25	Parallel mode SPI slave clock in, connecting to MCU side (upstream SPI)
Sck_S_L	I	24	Serial mode SPI slave clock , connecting to MCU side (upstream SPI)
Cs_L	I	23	Parallel mode SPI slave chip select signal , connecting to MCU side (upstream SPI)
Cs_S_L	I	22	Serial mode SPI slave chip select signal , connecting to MCU side (upstream SPI)
Sdo_L	O	21	SPI slave serial data out , connecting to MCU side (upstream SPI)
Sdi_L	I	20	Parallel mode SPI slave serial data in , connecting to MCU side (upstream SPI)
Sdi_S_L	I	19	Serial mode SPI slave serial data in , connecting to MCU side (upstream SPI)
Test_En	I	17	Test mode enable 1 for test mode and 0 for normal mode
Sdo_R	O	9	SPI master serial data out , connecting to downstream SPI port
Sdi_S_R	I	8	Serial mode SPI master serial data in , connecting to downstream SPI port
Sdi_R	I	7	Parallel mode SPI master serial data in , connecting to downstream SPI port
Cs_R	O	6	SPI master chip select signal out , connecting to downstream SPI port
Sck_R	O	5	SPI master clock out , connecting to downstream SPI port

Resetn_R	O	3	Resetn out, connecting to downstream
Start_en_R	O	4	One Start enable out, connecting to downstream
IO_mux_R	I	11	IO mode selection (downstream) 0 for parallel mode and 1 for serial mode
IO_mux_L	I	12	IO mode selection (upstream) 0 for parallel mode and 1 for serial mode
Start_en_S_L	I	26	Serial mode One Start enable input, connecting to MCU side (upstream)
Start_en_L	I	27	Parallel mode One Start enable input, connecting to MCU side (upstream)
Resetn_S_L	I	28	Serial mode low active Resetn input, connecting to MCU side (upstream)
Resetn_L	I	29	Parallel mode low active Resetn input, connecting to MCU side (upstream)
Clock_L	I	14	Parallel mode Reference clock input, connecting to MCU side (upstream)
GPIO0	I/O	18	Used to observe the inner signal
GPIO1	I/O	10	Used to observe the inner signal
EFUSE_En	I	16	Efuse mode 1: efuse signals are from spi signals 0: efuse signals are from efuse ctrl module

3 PROTOCOL SPECIFICATION

3.1 GENERAL PROTOCOL SPECIFICATIONS

The A9 Zmaster uses two SPI ports, one used in slave mode as an input from the MCU or the previous A9 Zmaster in the chain (pins SDO_L, SDI_L, SCK_L, CS_L) and one in master mode to the next downstream A9 Zmaster in the chain (pins SDO_R, SDI_R, SCK_R, CS_R), if any.

The SPI bus can operate at any clock speed up to 20 MHz. The clock is supplied by the MCU through the SCK_L pin and passed to the next A9 Zmaster in the chain through SCK_R.

A 16-bit SPI bus is used all the time, every frame is made up of 16-bit command/address field followed by 16-bit based data, if any data is to be followed by the command. The CS pins are used to qualify one frame from the master. There is no need for CS on the response frames because the MCU has knowledge of the length that the frame should have.

At reset, an automatic configuration mode enumerates and assigns addresses to all the chips in the chain. The first one will get address 0x01, the second one 0x02 and so on up to the last chip in the chain or to the maximum possible address of 0xFD.

Most command will loop back to MCU through the chain to indicate to the MCU that the command was executed. Some command such as BIST_ON or AUTO_ADDRESSING command will modify the 0x00 address field. For example, the AUTO_ADDRESSING command returns the frame with the highest chip address in the chain back to the MCU.

IMPORTANT NOTICE

Please note that all data and specifications are subject to change without notice. All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.

INNOSILICON Technology Ltd.

www.INNOSILICON.com

sales@innosilicon.com