2. Zynq Schematic

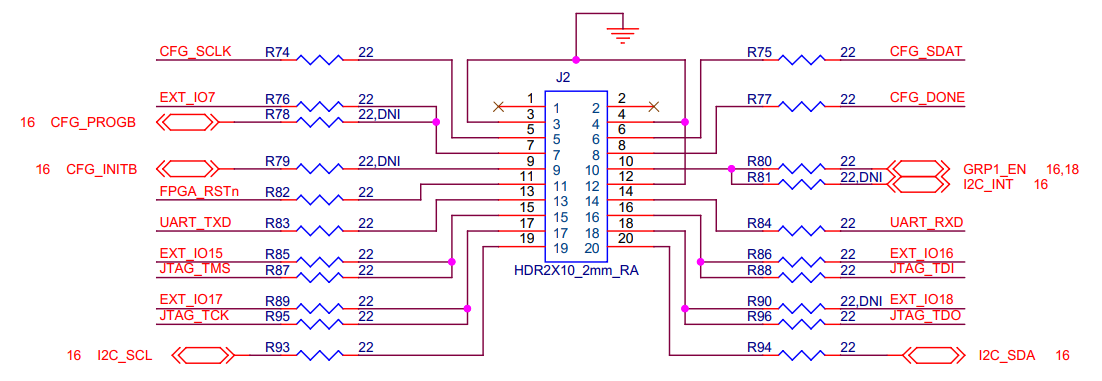
Zynq board is EBAZ4205. All public information could be found here: <https://github.com/xjtuecho/EBAZ4205>

3. MP board dsecription

3.1. Pins name and assignment: please view the constrain file constr\_e300.xdc

3.2. Pin assignment on 20-pin connector

Here is diagram of 20-pin connector:



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pin number | Pin name | Direction | Pin assignment on VU35P | Pin type on VU35P | Note |
| 1 | N/A |  | N/A | N/A |  |
| 2 | N/A |  | N/A | N/A |  |
| 3 | GND |  | N/A | N/A |  |
| 4 | GNA |  | N/A | N/A |  |
| 5 | CFG\_SCLK | in | BA6 | CCLK |  |
| 6 | CFG\_SDAT | in | AV7 | D01\_DIN |  |
| 7 | CFG\_PROGB | in | AW9 | PROGRAM\_B |  |
|  | EXT\_IO7 | in | AW14 | User IO |  |
| 8 | CFG\_DONE | out | BC6 | DONE |  |
| 9 | CFG\_INITB | out | BA7 | INIT\_B |  |
| 10 | GRP1\_EN |  | n/a | n/a | Set high level to this pin to power-up the MP board |
| 11 | FPGA\_RSTn | in | BE17 | User IO | Could use this pin as resetn signal for design |
| 12 | GND |  | n/a | n/a |  |
| 13 | UART TXD | in | C12 | User IO | This pin is for receiving uart signal from outside. The direction is INPUT. |
| 14 | UART RXD | out | B9 | User IO | This pin is for sending uart signal to outside. The direction is OUTPUT. |
| 15 | JTAG TMS | in | AY6 | JTAG dedicated pins |  |
|  | EXT\_IO15 | in | AV20 | User IO |  |
| 16 | JTAG TDI | in | AV6 | JTAG dedicated pins |  |
|  | EXT\_IO16 | in | AW19 | User IO |  |
| 17 | JTAG TCK | in | BB6 | JTAG dedicated pins |  |
|  | EXT\_IO17 | in | AW18 | User IO |  |
| 18 | JTAG TDO | out | AY7 | JTAG dedicated pins |  |
|  | EXT\_IO18 | out | AY17 | User IO |  |
| 19 | I2C SCL | inout | AY26 | User IO | The I2C signals connect to vu35p and other slave devices on board. The pin AY26 and AY27 is used when fpga is a slave. |
| 20 | I2C SDA | Inout | AY27 | User IO |

3.3. I2C slave devices

3.3.1. VU35P

The VU35P is one of i2c slave devices on board. When using vu35p as slave, developers have to implement i2c slave module and defined i2c addesss is 011\_0010. The pin assigment in this case is AY26 for SCL and AY27 for SDA.

3.3.2. TMP411C

This is a TI temperature sensor. It has 2 channels: local channel for monitoring board temperature, remote channel for monitoring vu35p chip temperature. The i2c address is 0x4E. Refer to TMP411 Datasheet for more information.

3.3.3. CDCE6214

This is a TI clock generator. Currently it has 4 outputs differential clock, all are 100Mhz. There is 100Ohm termination resistor on each clock path. To make the clock working properly, add the termination property in constrain like this:

set\_property PACKAGE\_PIN BB18 [get\_ports refclk\_2\_p]

set\_property IOSTANDARD LVDS [get\_ports refclk\_2\_p]

set\_property DIFF\_TERM\_ADV TERM\_100 [get\_ports refclk\_2\_p]

set\_property PACKAGE\_PIN BC18 [get\_ports refclk\_2\_n]

set\_property IOSTANDARD LVDS [get\_ports refclk\_2\_n]

set\_property DIFF\_TERM\_ADV TERM\_100 [get\_ports refclk\_2\_n]

The i2c address is 110\_0111 for Fallback mode and 110\_1000 for Normal mode.

Refer to constrain file for clock pin assignment.

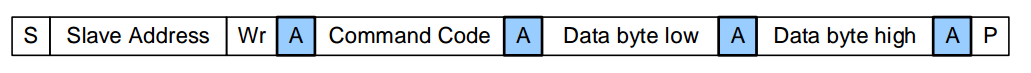
Refer to TI CDCE6214 Datasheet for more information.

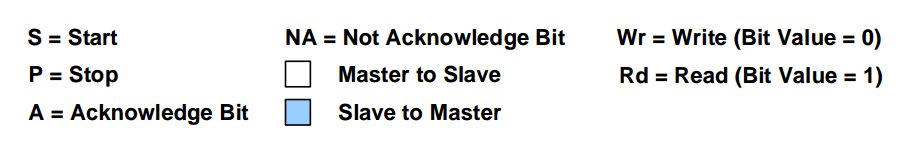
3.3.4. MP2975GU

This is power management chip. User can read or change value of VccINT and VccHBM. The i2c address is 0x7C.

Here is i2c message structure:

- for writing:

- for reading:



\*\*\*\* READ VOLTAGE \*\*\*\*\*

Step to read VccINT:

- Write command code 0x00 and one data byte 0x00

- Write command code 0x8B and read 2 data bytes. The first byte is low byte, the second byte is high byte.

- The returning voltage value is 850mV in default.

Step to read VccHBM:

- Write command code 0x00 and one data byte 0x01

- Write command code 0x8B and read 2 data bytes. The first byte is low byte, the second byte is high byte.

- The returning voltage value is 1265mV in default.

\*\*\*\*\* CHANGE VOLTAGE \*\*\*\*\*

The voltage can be changed between -128 → 127 steps, with each step at 5mV. The step to change voltage VccINT and VccHBM is below:

- Write command code 0x00 and one byte data 0x02

- Write command code 0x1E and two bytes data: byte 1 and byte 2, byte 1 is for VccINT, byte 2 is for VccHBM.

For example:

- to increase VccINT 20mV ↔ 4 steps ↔ byte 1 = 0x04

- to decrease VccHBM 20mV ↔ - 4 steps ↔ byte 2 = 0xFC

So the written sequence is: 0x1E 0x04 0xFC.