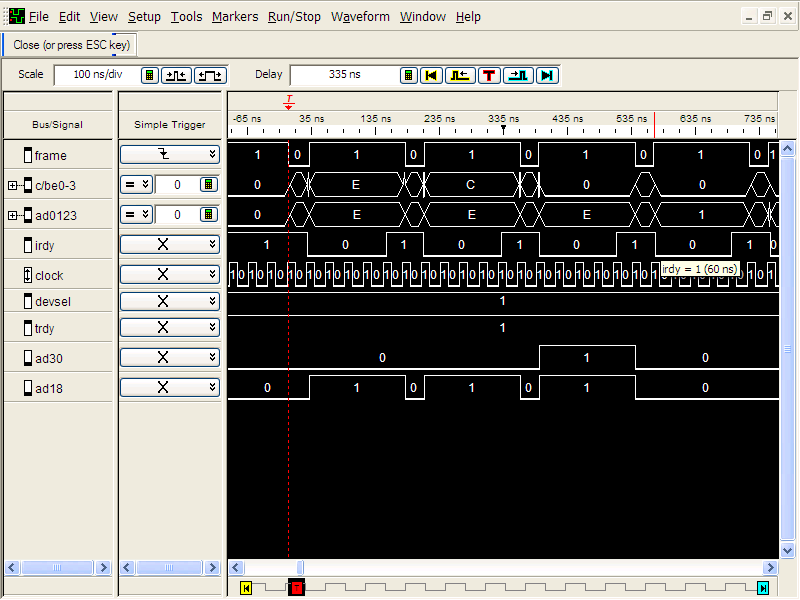
In my design TI PCI Bridge 2050BIZHK is used. The design is based on MPC 8548E processor on Integrity 5011 RTOS. The block diagram of configuration is described in figure below

In this design MPC 8548E is used as a processor and PCI Bridge is used to connect the PCI slave on other board via motherboard. PCI Bridge is connected on PCI 2 bus on primary side of processor.

Problem Description :  Device Vendor’s ID and Device ID are read properly but, read and write to the device is not working. Snapshot of the different control and data signals are as below (Write cycle)



Data (AD0-AD3) snapshot is only for last four bit of data (LSB).

The bridge configuration details are as follows: (BusRead4B reads 4 bytes data on PCI Bus and BusWrite4B writes 4 bytes of data, command is 4 bytes variable)

command = BusRead4B(configSpace, configHandle, 0x00); /\*Reading Vendor ID & Device ID\*/

command = BusRead4B(configSpace, configHandle, 0x04); /\* Reading Command & Status Word\*/

command = 0x02900347;

BusWrite4B (configSpace, configHandle, 0x04, command); /\* Modifing Command & Status Word- I/O Space Enable, Memory space enable, Bus master Capability enable, Parity Error response enable, System error enabled\*/

command = BusRead4B(configSpace, configHandle, 0x08); /\*Reading revision ID (0x02) and Class Code (0x060400) \*/

command = 0x00010808;

BusWrite4B (configSpace, configHandle, 0x0c, command); /\* Modifing Catch line size+Primary latency+Header Type+BIST Cache line size set to 0x08, Primary Latency timer set to zero clock count reading header hypr as 0x01 and BIST as 0x00\*/

command = BusRead4B(configSpace, configHandle, 0x10); /\*Reading Base Address Register 0 as 0x00000000\*/

command = BusRead4B(configSpace, configHandle, 0x14); /\*Reading Base Address Register 1 as 0x00000000\*/

command = 0x0100;

BusWrite2B (configSpace, configHandle, 0x18, command);/\* Assigning Primary Bus Number as 0 and Secondary bus number as 1\*/

command = 0x0001;

BusWrite2B (configSpace, configHandle, 0x1a, command); /\*Assigning Subordinate Bus Number as 1 and secondary bus latency timer as 0 clock count \*/

command = 0x2280f1f1;

BusWrite4B (configSpace, configHandle, 0x1c, command); /\*Setting Io Base register(AD 15-AD 12) as 0xf and IO limit register oxf\*/

command = 0x8000;

BusWrite2B (configSpace, configHandle, 0x20, command); /\*AD32 - AD 20 has memory base value of 0x8000\*/

command = 0x9ff0;

BusWrite2B (configSpace, configHandle, 0x22, command); /\*AD32 - AD 20 has base memory limit value 0x9ff0\*/

command = 0x8000;

BusWrite2B (configSpace, configHandle, 0x24, command); /\*Same as memory base value of 0x8000\*/

command = 0x9ff0;

BusWrite2B (configSpace, configHandle, 0x26, command); /\*Same as memory base limit value 0x9ff0\*/

command = 0x00000000;

BusWrite4B (configSpace, configHandle, 0x28, command); /\*As 32 bit operation\*/

command = 0x00000000;

BusWrite4B (configSpace, configHandle, 0x2c, command); /\*As 32 bit operation\*/

command = 0xffffffff;

BusWrite4B (configSpace, configHandle, 0x30, command); /\*IO Base and IO Upper limit AD31-AD16\*/

command = BusRead4B(configSpace, configHandle, 0x34);/\*Capabitity pointer register read as 0xdc\*/

command = BusRead4B(configSpace, configHandle, 0x38); /\* Expansion ROM Base address read as 0x0000\*/

command = BusRead2B(configSpace, configHandle, 0x3e); /\* BRIDGE CTRL: read as 0x0000\*/

command = BusRead2B(configSpace, configHandle, 0xE0); /\* PWR MNGT CTRL: read as 0x0000\*/

command = BusRead2B(configSpace, configHandle, 0xE4);/\* Capability ID register read as 0x00- Intel Compatible mode\*/

command = 0x0010;

BusWrite2B (configSpace, configHandle, 0x40, command);/\* Enable Memory Read Prefetch BIT-4 \*/