DIAMOND SYSTEMS CORPORATION

Helios Downclocking

Technical Note

Revision: June 2010

Revision	Date	Comment
A	6/10/2010	Initial version

FOR TECHNICAL SUPPORT PLEASE CONTACT:

support@diamondsystems.com

Diamond Systems Corporation
1255 Terra Bella Ave.
Mountain View, CA 94043 USA
Tel 1-650-810-2500
Fax 1-650-810-2525
www.diamondsystems.com

Change CPU Speed

Internally the Vortex86SX/DX usesPLL (Phase-Locked Loop) technology, which is tied to the CPU clock and can be adjusted by changing the register value of the North Bridge Offset register A0h.

The CPU speed could be divided from 1 to 8 by the default CPU clock. For example, if the default CPU clock is 300MHz, and you choose"CPU speed divide by 5", the CPU speed will be 300 / 5 = 60 MHz.

MINIMUM SPEED

The CPU speed cannot be lower than the PCI speed which is 33MHz.

To change CPU clock, find PCI register A0H in north bridge (Vendor ID: 17F3, Device: 6021). Bit 7-3 is reserved and bit 2-0 is CPU speed divided control:

Bit[2-0]	Description
000	Divide 1
001	Divide 2
010	Divide 3
011	Divide 4
100	Divide 5
101	Divide 6
110	Divide 7
111	Divide 8

For example: if CPU clock is 300MHz and set speed divided control to "001", CPU clock will be 150MHz.

Assembler Example

```
mov dx,  cf8h ; PCI address port set = north bridge offset register a0h
mov eax, 800000a0h
out dx,  eax

mov dx,  cfch ; PCI data port read / write
in  eax, dx
                ; if CPU clock is  300MHz or
eax, 00000001h ; set CPU clock to 150MHz or
eax, 00000004h ; set CPU clock to  60MHz out
dx,  eax
```

DOS Example

```
#include <stdio.h>
#include <dos.h>

// Disable warning message "Parameter xxx is never used
#pragma warn -par

// Read north bridge register
unsigned long read_nb(unsigned char idx)
{
  unsigned long retval;
  _asm mov dx, 0cf8h
  __emit__(0x66); __emit__(0xb8);
  __emit__(0x00); __emit__(0x00); __emit__(0x00); __emit__(0x80);
  _asm mov al, idx
  __emit__(0x66); _asm out dx, ax
  _asm mov dx, 0cfch
  __emit__(0x66); _asm in ax, dx
  __emit__(0x66); _asm mov WORD PTR retval, ax
  return retval;
}

// Write north bridge register
void write_nb(unsigned char idx, unsigned long val)
{
  _asm mov dx, 0cf8h
  __emit__(0x66); __emit__(0xb8);
  __emit__(0x00); __emit__(0x00); __emit__(0x00); __emit__(0x80);
  _asm mov al, idx
  __emit__(0x66); __emit__(0xef);
  _asm mov dx, 0cfch
  __emit__(0x66); __emit__(0x8b); __emit__(0x46); __emit__(0x08);
  __emit__(0x66); __emit__(0xef);
                }

void main()
     {
  unsigned char c;

  /* Read A0H in north bridge (Vendor ID: 17F3, Device: 6021).
     Bit 7-3 is reserved and bit 2-0 is CPU speed divided control. */
  c = read_nb(0xA0);
  c &= ~0x07;  /* Clear bit 2-0 */

  /* Set clock: bit[2-0]
     000 -> Divide 1
     001 -> Divide 2
     010 -> Divide 3
     011 -> Divide 4
     100 -> Divide 5
     101 -> Divide 6
     110 -> Divide 7
     111 -> Divide 8 */
  c |= 0x01;  /* If CPU is 300MHz, set clock to 150MHz */

  write_nb(0xA0, c);
         }
```