

Features

- Use the Windows Calculator as a tool for determining register values
- Calculate Range, Chip Select, and other register values
- Includes a table of powers of 2

General Description

This application note describes simple methods for calculating values to program into Local Range, Chip Select and other registers, using the Windows calculator as a tool. This application note describes, for example, how to determine the Range register value for a required address space size, with one click of the mouse.



Figure 1. Windows Calculator in Scientific Method Mode

Contents

1. Introduction	2
1.1 Calculator operation	2
1.2 Calculator memory	2
2. Local Range Register	2
3. Local Base Address Register	3
4. Chip Select Register	4
5. Other Registers	4
6. Table of Powers of 2	5

Figures

Figure 1. Windows Calculator in Scientific Method Mode	1
--	---

1. Introduction

- 1) Open the Windows calculator (Start/Programs/Accessories/Calculator).
- 2) Select the View Scientific option (as shown above), if the Standard calculator is displayed.
- 3) Select the Hex (hexadecimal) radio button, or press the F5 key. The F6 key selects decimal base.
- 4) Select the Dword radio button, or press the F2 key.

1.1 Calculator operation

Calculator functionality should generally be straightforward and intuitive for most operations. Numbers (0-1, 0-9, or 0-9 & A-F, as appropriate for the selected base) can be entered directly, or as the result of an operation, using the mouse or keyboard.

For example, to enter 64KB using the mouse,

- in Dec base click **65536**, or **2 x^y 16**, or **1 Lsh 16**
- in Hex base click **10000**, or **2 x^y 10**, or **1 Lsh 10**

The keystroke equivalent sequence for the above example in hexadecimal base is: **10000**, or **2 y 10**, or **1 < 10**

- Values can be shifted left, as indicated in the example above, with the **Lsh** key.
- Values can be shifted right one or more bits at a time by dividing by 2 or a power of two. (For example, shift 4 bits to the right by dividing by 10 in Hex base).

To learn more about the calculator, click **Help**, or key **ALT-H**.

1.2 Calculator memory

One or more displayed values can be stored in calculator memory. A single value can be stored or recalled by using the **MS** and **MR** buttons, respectively, or the respective keyboard equivalents, **CTRL+M** and **CTRL+R**. Values may also be imported or exported by using the Windows clipboard **Copy** and **Paste** commands in the **Edit** menu.

Multiple values can be stored by using statistical functions memory. To use statistics memory, open the Statistics Box by clicking the **STA** button, or key **CTRL+S**. One or more values can then be copied from the display to the Statistics Box by clicking the **DAT** button, or by keying **INS**. A value can be copied back from statistics memory to the display, by first bringing the Statistics Box into focus by clicking the **STA** button or by keying **CTRL+S**, and then by highlighting the entry and either clicking the **Load** key, or by keying **ALT-L**.

The Statistics Box must be kept open if values are to be preserved. However, the Statistics Box can be moved to the background, with focus restored to the calculator, by clicking the **Ret** button or by keying **ALT-R**, or by pressing the **Enter** key if the **Ret** button is marked as the default action button. The **CD** and **CAD** buttons clear one or all values from statistics memory.

2. Local Range Register

To calculate the Local Range register value (for the size of the address space):

1. Enter the size of the address space as a power of two.
 - For memory space, minimum size is 16 (10h) bytes.
 - For I/O space, minimum size is 4 bytes. I/O should not exceed 256 (100h) bytes per address space, per the PCI specification.
 - If the value is known and it is a power of two, select the appropriate radio button for Hex, Dec, or Bin base, then enter the value.
 - To otherwise calculate the value, it can be handy to memorize a few conversions, such as 1KB = 1024 = 2¹⁰ and 1MB = 1024 x 1024 = 2²⁰
 - A table is included at the bottom of the page.

2. Verify the value is a power of two by selecting the Bin radio button and seeing that only one bit is set to 1, with all other bits 0. Then select Hex base.
3. The size can be saved in calculator memory (for calculating the chip select value, below).
4. Click the +/- button (to the right of 0), or press the **F9** key. For example, with 64KB displayed as 10000, clicking +/- (or pressing the F9 key) should display FFFF0000.
 - If the display shows a minus sign (-10000 for the example above), click the = button.

The +/- button (or F9 key) converts the address space size value to its two's complement, which is the value to be programmed into the Range register bits [31:4] for memory, or bits [31:2] for I/O space. For I/O, bits [31:8] (upper three bytes) should be all 1s.

5. Additionally
 - For memory space
 - Bit 0 must be set to 0.
 - Bits [2:1] should be set to 00 to allow BIOS to map the space anywhere within 32-bit PCI address space. On the PC platform address space below 1MB is reserved for system use for platform compatibility.
 - Bit 3 should be set to 1 if prefetch is enabled in the Local Address Space Descriptor register, otherwise set to 0, to indicate prefetch capability status to the system. BIOS will copy this status bit-to-bit 3 of the corresponding PCIBAR register.
 - For I/O space,
 - Bits [1:0] must be set to 01b
 - Prefetch is not used.

BIOS will utilize the Range register values for allocation of PCI address space. During the boot process, BIOS will write all 1's to a PCIBAR register, and then read the register. This writing of all 1's to a PCIBAR register signals the PLX chip to return the value in the Local Range Register when the PCIBAR is read, providing BIOS with the size and type of address space required. The BIOS then (normally) allocates PCI address space, and writes the base address for this address space into the PCIBAR register.

3. Local Base Address Register

The Local Base Address register contains the local base address, with bit 0 set to 1 to enable the space, and bit 1 clear. The base address must be a multiple of the size. For example, with 1MB displayed as 100000 in Hex base, the base address could be 0, 1MB, 2MB (200000h), 3MB (300000h) and so on. The base address will be on a 16-byte boundary for memory space or on a 4-byte boundary for I/O space.

If the address space is I/O on the PC platform, the entire address space should lie within the lowest 256 bytes of one of 63 1K ranges (the first 1K (0-3FFh) is reserved for system use).

Therefore for I/O on the PC platform, bits [11:10] (C00h) should be zero for any address within the space, and bits [31:16] must be zero.

1. Enter the base address.
2. Set bit 0 to one by keying **Or 1 =**, to enable the address space.
3. If this base address is also to be used for a Chip Select, save this value in calculator memory for future use.

4. Chip Select Register

The Chip Select register value encodes both the Local Base Address and address space size.

- The size encoding is one bit, calculated by shifting the size value, which must be a power of two, one bit to the right.
- The base address is defined in the bits to the left of the size bit. Bits [31:28] are always zero.
- Bit 0 being set to 1 enables chip select decoding.
 1. Enter the size of the address space, as was done above for the range register, or retrieve the value from calculator memory, if the size value was previously saved as described above. For example, for the minimum 4 bytes of I/O, key in 4.
 2. Shift the value right one bit by dividing by 2 (Hex or Dec base).
 3. Add or OR a base address value to any bits to the left (as seen in Hex or Bin base). This base address value will be the same as used for the Local Base Address register. For example, in Hex base with 1MB the shifted right value will be 00080000. For a base address of 1MB (100000h), click **Or (1 Lsh 14) =** which results in the value 00180000. The 14 is hex base. Another example, with 4 bytes (I/O) the size encoding will be 00000002. For a base address of 400h, click **Or 400 =** which results in value 00000402.
 4. OR a 1 to bit 0 of the value by keying **Or 1 =**, to enable the Chip Select. If bit 0 remains set to 1 from the base address calculation above, this step can be skipped. For the preceding example, the resulting Hex value will be 00000403.

The hexadecimal value displayed is the Chip Select register value.

5. Other Registers

The calculator can be used to generate values for other registers.

- A bit can be set to 1 using the sequence **Or (1 Lsh x) =** where x is a bit number. In Hex base, x is a hexadecimal value 0 - 1F (decimal 0 - 31).
- A bit can be cleared using the sequence **And (1 Lsh x) Not =** . If the bit is known to be 1, you can use the sequence **Xor (1 Lsh x) =** .
- A higher number for multiple bit manipulations can replace the 1 in the examples above. For example, to set bits [22:19] to a value of 4 in a number currently displayed in Hex base, click **And (F Lshf 13) Not Or (4 Lsh 13) =**.
- The keyboard equivalent for the above example is: **& (F < 13) ~ | (4 < 13) ENTER.**

6. Table of Powers of 2

Power	Decimal	Hex
2 ⁰	1	1
2 ¹	2	2
2 ²	4	4
2 ³	8	8
2 ⁴	16	10
2 ⁵	32	20
2 ⁶	64	40
2 ⁷	128	80
2 ⁸	256	100
2 ⁹	512	200
2 ¹⁰	1024	400
2 ¹¹	2028	800
2 ¹²	4096	1000
2 ¹³	8192	2000
2 ¹⁴	16384	4000
2 ¹⁵	32768	8000

Power	Decimal	Hex
2 ¹⁶	65536	10000
2 ¹⁷	128K	20000
2 ¹⁸	256K	40000
2 ¹⁹	512K	80000
2 ²⁰	1M	100000
2 ²¹	2M	200000
2 ²²	4M	400000
2 ²³	8M	800000
2 ²⁴	16M	1000000
2 ²⁵	32M	2000000
2 ²⁶	64M	4000000
2 ²⁷	128M	8000000
2 ²⁸	256M	10000000
2 ²⁹	512M	20000000
2 ³⁰	1G	40000000
2 ³¹	2G	80000000