Clearing Memory with the 32000; Series 32000® Graphics Note 3

National Semiconductor
Application Note 527
Dave Rand
May 1988

1.0 INTRODUCTION

In printer applications, large amounts of RAM may need to be initialized to a zero value. This application note describes a fast method.

2.0 DESCRIPTION

While several different methods of initializing memory to all zeros are available, here is one that works very well on the Series 32000. While the current version clears memory only in blocks of 128 bytes, other block sizes are possible by extending the algorithm.

3.0 IMPLEMENTATION

This routine is written to clear blocks of 128 bytes. This provides an optimal tradeoff between loop size (granularity) and loop overhead. This can be modified to use a different size. For example, to use a block size of 64 bytes, simply delete 16 of the MOVQD 0,TOS instructions from the listing. As well, since the value of r1 is now the number of 64 byte groups, one of the ADDD R2,R2 instructions (prior to the loading of the stack pointer) must be removed. Since the 32000 has two stacks, interrupts will be handled properly using this code. If only a fixed buffer size needs to be cleared, the code can be further unrolled to clear that area (i.e., increase the number of MOVQD 0,TOS instructions.)

```
; Version 1.1  Sun Mar 29 10:22:19 1987
;
;Subroutine to clear a block of memory. The granularity of this
;algorithm is 128 bytes, to reduce the looping overhead.
;
;       Inputs:
;               r0 = start of block
;               r1 = number of 128-byte groups to clear
;
;       Outputs:
;               All registers preserved.
;
;
;Listing continues on following page
;
```

TL/EE/9697–1

Series 32000® is a registered trademark of National Semiconductor Corporation.

RRD-B30M105/Printed in U. S. A.

```
clram:  cmpqd   0,r1            ;any blocks to clear?
        beq     clexit:w        ;no, exit now.
        save    [r0,r1,r2]      ;save our working registers
        movd    r1,r2           ;here we set r0 = r0 + (r1 * 128) + 4
        addd    r2,r2           ;length *= 2
        addd    r2,r2           ;*4
        addd    r2,r2           ;*8
        addd    r2,r2           ;*16
        addr    4(r0)[r2:q],r0  ;get starting point + 4
        sprd    sp,r2           ;save current stack
        lprd    sp,r0           ;move to last double
        .align  4
cl2:    movqd   0,tos           ;clear a double
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        movqd   0,tos
        acbd    -1,r1,cl2
        lprd    sp,r2         ;restore stack pointer
        restore [r0,r1,r2]    ;restore our saved registers
clexit: ret     0
```

TL/EE/9697–2

FIGURE 1

```
clram:  cmpqd   0,r1          ;any blocks to clear?
        beq     clexit:w      ;no, exit now.
        .align  4
cl2:    movqd   0,00(r0)            ;clear a double
        movqd   0,04(r0)
        movqd   0,08(r0)
        movqd   0,12(r0)
        movqd   0,16(r0)
        movqd   0,20(r0)
        movqd   0,24(r0)
        movqd   0,28(r0)
        movqd   0,32(r0)
        movqd   0,36(r0)
        movqd   0,40(r0)
        movqd   0,44(r0)
        movqd   0,48(r0)
        movqd   0,52(r0)
        movqd   0,56(r0)
        movqd   0,60(r0)
        movqd   0,64(r0)
        movqd   0,68(r0)
        movqd   0,72(r0)
        movqd   0,76(r0)
        movqd   0,80(r0)
        movqd   0,84(r0)
        movqd   0,88(r0)
        movqd   0,92(r0)
        movqd   0,96(r0)
        movqd   0,100(r0)
        movqd   0,104(r0)
        movqd   0,108(r0)
        movqd   0,112(r0)
        movqd   0,116(r0)
        movqd   0,120(r0)
        movqd   0,124(r0)
        addd    $128,r0
        acbd    -1,r1,cl2
clexit: ret     0
```

TL/EE/9697–3

FIGURE 2

4.0 TIMING RESULTS

On the NS32016, NS32032 and NS32332, 4 clock cycles per write are required. To clear one page of 300 DPI 8½ x 11 (1,056,000 bytes), for example, requires 264,000 double words to be written. The optimal time for this, using 100% of the bus bandwidth on a 16 bit bus, would be 528,000 * 400 ns, or 211.2 ms, @ 10 MHz. All timing data assumes word aligned data (double word aligned for 32 bit bus). Unaligned data is permitted, but will reduce the speed somewhat.

On the NS32332 (no wait states. @15 MHz, 32 bit bus), this code clears the full page image in 178 ms.

On the NS32032 (no wait states. @10 MHz, 32 bit bus), this code clears the full page image in 324 ms.

On the NS32016 (1 wait state. @10 MHz, 16 bit bus), this code clears the full page image in 509 ms.

Doubling the block size (to 256 bytes) would increase the speed by 1%–2%, on the code sample.

On the NS32532, a better approach is to use the register indirect method of referencing memory, as is shown in *Figure 2*. With this approach, the page memory can be cleared in 19 ms, assuming a no wait state 30 MHz system, with a 32 bit bus. The optimal time, using 100% of the bus bandwidth of the NS32532 (2 clock bus cycle) would be 264,000 * 66.6 ns, or 17.6 ms.

Lit. # 100527

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408