Implementing a 1K Byte RAM on the NS32GX320

National Semiconductor
Application Note 763
Zeev Bikowsky
March 1991

INTRODUCTION

Many one-chip microcontrollers have an on-chip internal RAM (typically 128–256 bytes). Internal RAM reduces the time and bus load to access an external memory. The NS32GX320 has a 1,024 byte Data Cache, and a 512 byte Instruction Cache. This application note describes how to convert the Data Cache of the NS32GX320 into 1,024 bytes of internal RAM.

BACKGROUND

The NS32GX320 contains two on-chip caches: the Instruction Cache (IC) and the Data Cache (DC). These are used to store the contents of frequently used memory locations. The IC and the DC can be individually enabled by setting appropriate bits in the CFG register (see *Figure 1*). The NS32GX320 also provides a locking feature that allows the contents of the IC and the DC to be locked to specific memory locations. This is accomplished by setting the LIC and the LDC bits in the CFG register, refer to AN-608, ''Locking Code in the NS32GX32 Instruction Cache''.

Cache locking can be successfully used in real-time applications to guarantee fast access to critical instructions and data areas (this application note refers to the Data Cache only). Note that accesses to the on-chip peripherals are not cachable.

res	LIC	IC	LDC	DC	DE	1	1	1	1	C	res	F	I

FIGURE 1. Configuration Register (CFG)

DC (Data Cache enable) Enables the on-chip Data Cache.

LDC (Lock Data Cache) Controls whether the contents of the on-chip Data Cache are locked to fixed memory locations (LDC = 1), or updated when a data read is missing from the cache (LDC = 0).

IC (Instruction Cache enable) Enables the on-chip Instruction cache.

LIC (Lock Instruction Cache) Controls whether the contents of the on-chip instruction Cache are locked to fixed memory locations (LIC = 1), or updated when an instruction read is missing from the cache (LIC = 0).

DESCRIPTION OF THE DATA CACHE

The Data Cache (DC) stores 1,024 bytes of data in a two-way set associative organization. Each of the 32 sets has 2 cache blocks. Each block has a 23-bit tag, which holds the most-significant bits of the address of the locations stored in the block, along with 4 double-words and 4 validity bits (one for each double word). The DC is enabled for a data read when the two conditions are satisfied:

1. The DC bit in the CFG register is set to 1.
2. The reference is not an interlocked read resulting from executing a CBITI or SBITI instruction.

If the DC is disabled, it is bypassed during the data read and its contents is therefore unaffected. The data is read directly from external memory. The DC is also bypassed for Interrupt-Acknowledge and End-of-Interrupt bus cycles.

When the DC is enabled for a data read, the address bits 4 to 8 are used to select the DC set where the data may be stored.

The tags corresponding to the two blocks in the set are compared to the 23 most-significant bits of the address. Bits 2 and 3 of the address select one double-word in each block and the associated validity bit. If one of the tag matches and the selected double-word in the corresponding block is valid, a cache ''hit'' occurs and the data is used to execute the instruction; otherwise a cache ''miss'' will result. In the latter case, if the cache is not locked, the CPU will take the following actions.

First, if the tag of either block in the set matches the data address, that block is selected for updating. Otherwise, if neither tag matches, then the least recently used block is selected; its tag is loaded with the 23 most-significant bits of the data address, and all the validity bits are cleared.

Then the data is read from the external memory; up to 4 double-word bits are read into the cache in a wrap-around fashion.

If the CIIN and the $\overline{\text{IODEC}}$ output signals are both inactive during the bus cycles performed to read the missing data, then the DC is updated, as each double-word is read from memory, and the corresponding validity bit is set. If the cache is locked, its contents are not affected, as the CPU reads the missing data from external memory.

The DC is enabled for a data write whenever the DC bit in the CFG register is set to 1, including interlocked writes resulting from executing the CBITI and SBITI instructions. The DC does not use write allocation. This means that, during a write, if a cache ''hit'' occurs, the DC is updated, otherwise it is unaffected. The data is always written through the external memory.

The content of the data cache can be invalidated by software through the CINV instruction. Clearing the DC bit in the CFG register also invalidates the DC. The DC can be invalidated by activating the $\overline{\text{CINV}}$ pin.

Note: If the DC is enabled for a certain data reference and a ''miss'' occurs due to tag mismatch, the CPU will clear all the validity bits for the least recently used tag before reading the data from external memory. If either CIIN or $\overline{\text{IODEC}}$ are activated during the data read bus cycles, the validity bits are not set and the DC is not updated.

IMPLEMENTATION

Read data from 1,024 successive byte-addresses. Make sure that:

1. These 1024 byte addresses are not used for other purposes in the system.
2. The CIIN signal is 0 and the $\overline{\text{IODEC}}$ signal is 1.

The DC entries will be filled with undefined data by these read operations. At this point, lock the DC. Now run a second loop, to write initial values into these cache lines and from now on 1,024 bytes of RAM are on-chip. Initialization of the cache is not necessary and may be avoided.

Note that the cache in the NS32GX320 is write-through, thus, any writes to the on-chip RAM will go off-chip.

 TL/EE11184

RRD-B30M75/Printed in U. S. A.

Following is the software that implements the RAM setting:

```
.set start_address, h'00001000      # ram will start at address h'1000
                                    # ram start address can be h'xxxxx000

.text
start:                              #

     lprd cfg,      $h'bf3          # initialize CFG register,
                                    # caches on,38l on.
     movd $start_address, r0        # initialize start address.
     movd $256,rl                   # initialize # of double words
                                    # to load.
readloop:                           #

     movd 0(r0),r2                  # dummy-read, to load double
                                    # words into the cache
     addqd $4,r0                    # increment memory pointer by 4 (bytes)
     acbd $-l,rl,readloop           # loop count down
                                    # loop control statement
                                    # jump to next loop iteration/continue


     lprd cfg,      ,$h'ff3         # lock the data cache.

init:                               # initialize values to cache

     movd $start_address, r0        # initialize start address
     movd $256,rl                   # initialize # of double words
                                    # to load.
     movd $0,r7                     # value to load into the cache.

writeloop:                          # write to data cache

     movd r7,0(r0)                  # load r7 into the cache
     addqd $4,r0                    # increment memory pointer by 4 (bytes)
     acbd $-l,rl,writeloop          # loop count down
                                    # loop control statement
                                    # jump to next loop iteration/continue
```


National Semiconductor Corporation
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: 1(800) 272-9959
TWX: (910) 339-9240

National Semiconductor GmbH
Livry-Gargan-Str. 10
D-82256 Fürstenfeldbruck
Germany
Tel: (81-41) 35-0
Telex: 527649
Fax: (81-41) 35-1

National Semiconductor Japan Ltd.
Sumitomo Chemical
Engineering Center
Bldg. 7F
1-7-1, Nakase, Mihama-Ku
Chiba-City,
Ciba Prefecture 261
Tel: (043) 299-2300
Fax: (043) 299-2500

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 Semiconductores Do Brazil Ltda.
Rue Deputado Lacorda Franco
120-3A
Sao Paulo-SP
Brazil 05418-000
Tel: (55-11) 212-5066
Telex: 391-1131931 NSBR BR
Fax: (55-11) 212-1181

National Semiconductor (Australia) Pty, Ltd.
Building 16
Business Park Drive
Monash Business Park
Nottinghill, Melbourne
Victoria 3168 Australia
Tel: (3) 558-9999
Fax: (3) 558-9998