SPRZ125
April 1998

TMS320C82 Errata Sheet
December 31, 1997 – Version 1.0h

NOTE: Signal names beginning with the following character '/' indicate an active-low signal. Example: **/LINT4** is active low.

Identifying the TMS320C82 Device Revision:

The revision number for each TMS320C82 device can be determined by the device symbolization printed on the top of the device package. The device revision is identified by the first two numbers in the second line of the device symbolization. The example below indicates a Rev 1.1 device.

TMS320C82GGP
C119631
LOT # xxxxxxxx

Device Speeds:

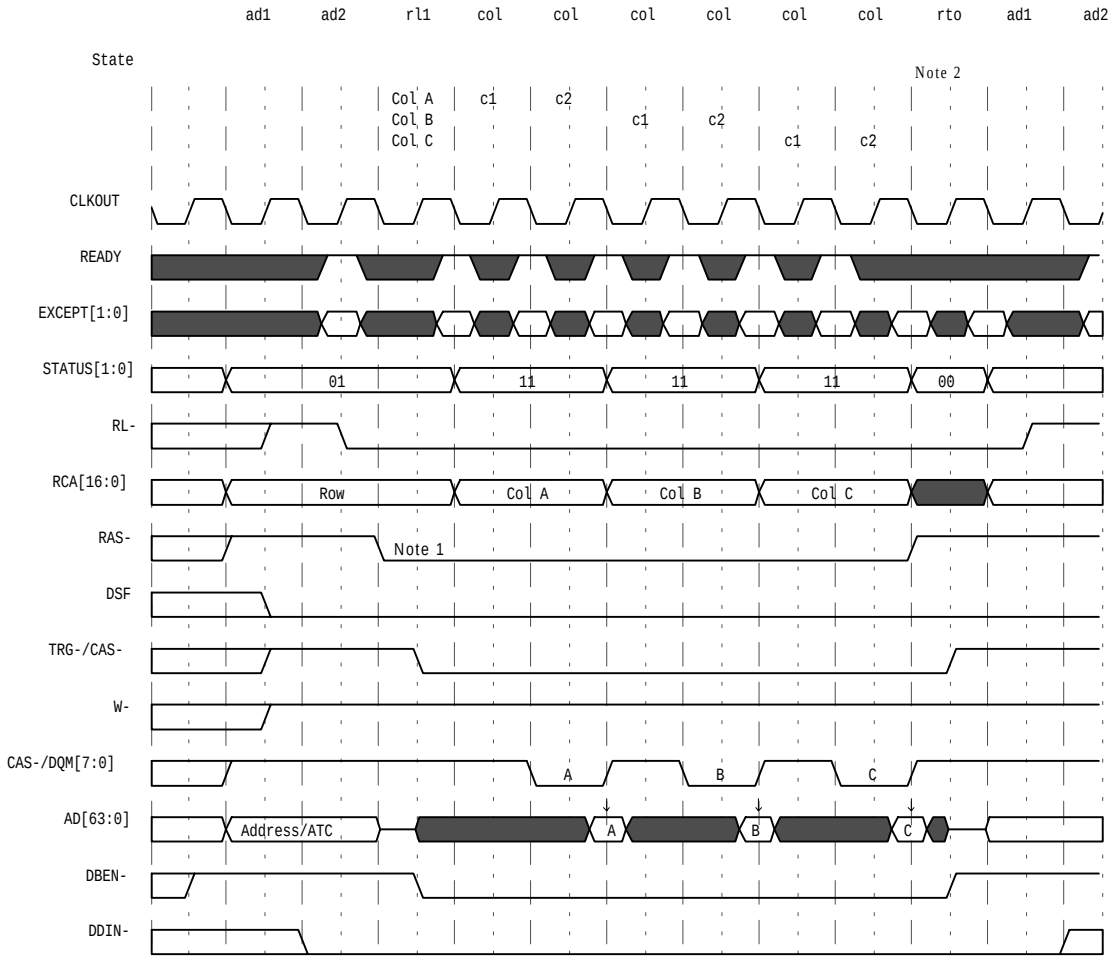
- Revision 1.0/1.1 devices are limited to 50 MHz operation (100 MHz CLKIN) (see Section 1.5). Parameter $t_{c(CKI)}$ must be at least 20 ns.

Silicon 1.x Device Errata:

1.1 2 cyc/col SRAM implementation:

The 2 cycle/column SRAM (CT = 0110) cycles are not implemented correctly. For 2 cycle/column reads, the fall of **/CAS** is delayed by ½ cycle. The following cycle diagram illustrates the operation of revision 1.0/1.1 silicon for 2 cycle/column SRAM reads.

2 Cyc/Col SRAM Read - Rev 1.0/1.1



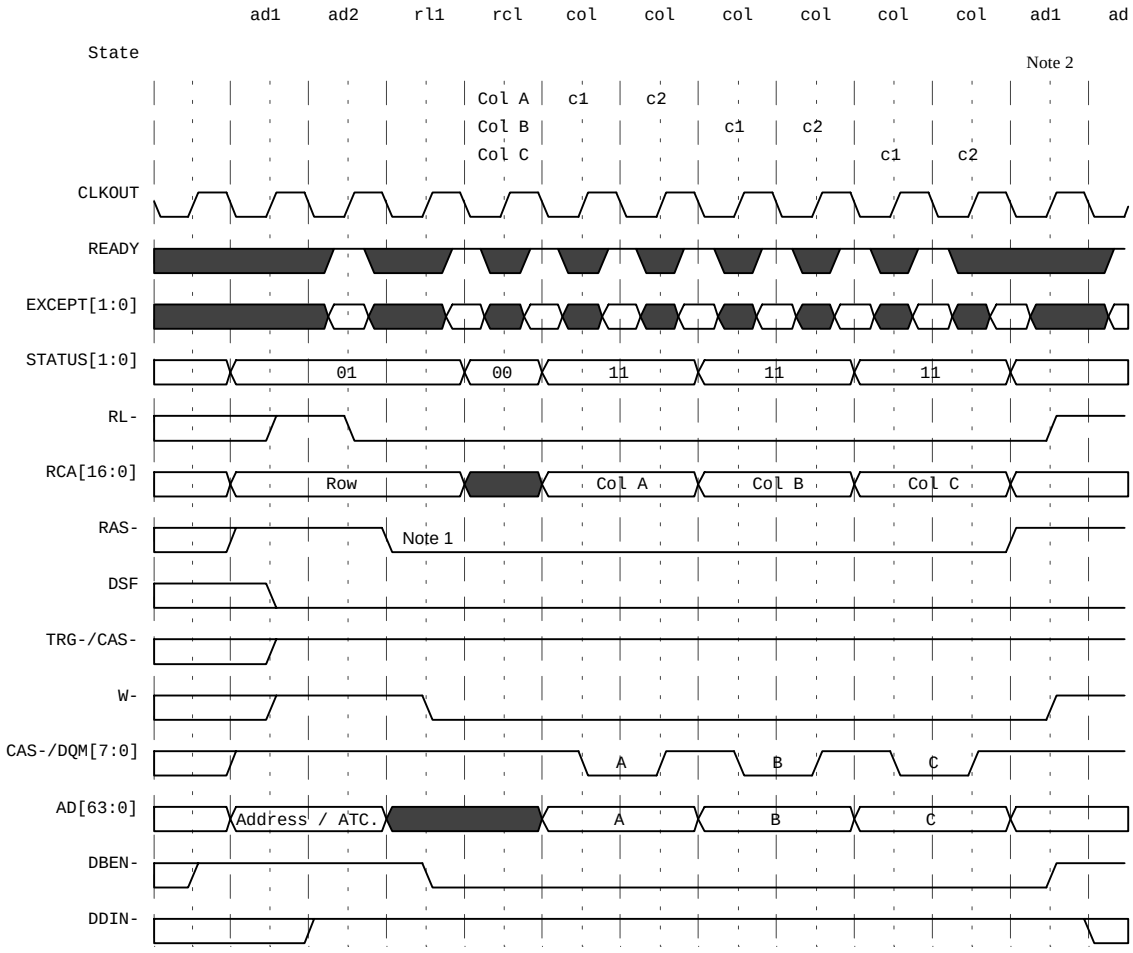
Note 1: No RAS- high time requirements are applied to these cycles.

Note 2: Additional turn-off cycles will be inserted between rto and ad1 as specified by the bank configuration.

1.1 2 cyc/col SRAM implementation: (continued)

In the revision 1.0/1.1 silicon, the 2 cycle/column SRAM write cycles are implemented as shown below. Note that the fall of **/CAS** is delayed by ½ cycle.

2 Cyc/Col SRAM Write - Rev 1.0/1.1



Note 1: No RAS- high time requirements apply to these cycles.

Note 2: During peripheral data transfers, turn-off cycles will be inserted prior to ad1 as specified by the bank configuration.

Workaround:
None. As a consequence, SRAMs with faster access times may be required to interface at this cycle timing.

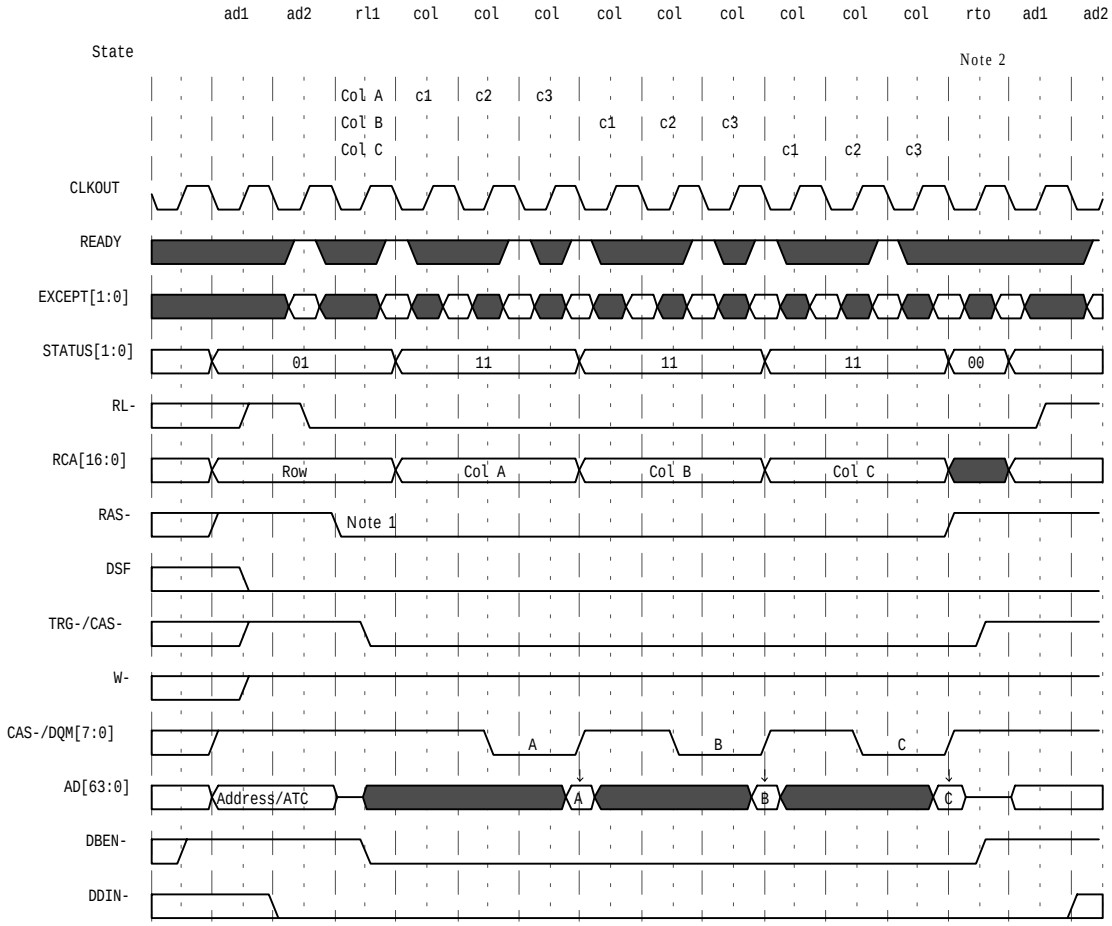
Problem exists for silicon revisions: 1.0, 1.1, 1.2

Fix verified for silicon revision: 1.3

1.2 3 cyc/col SRAM implementation:

The 3 cycle/column SRAM (CT = 0111) cycles are not implemented correctly. For 3 cycle/column reads, the fall of **/CAS** is delayed by ½ cycle. The following cycle diagram illustrates the operation of revision 1.x silicon for 3 cycle/column SRAM reads.

3 Cyc/Col SRAM Read - Rev 1.x



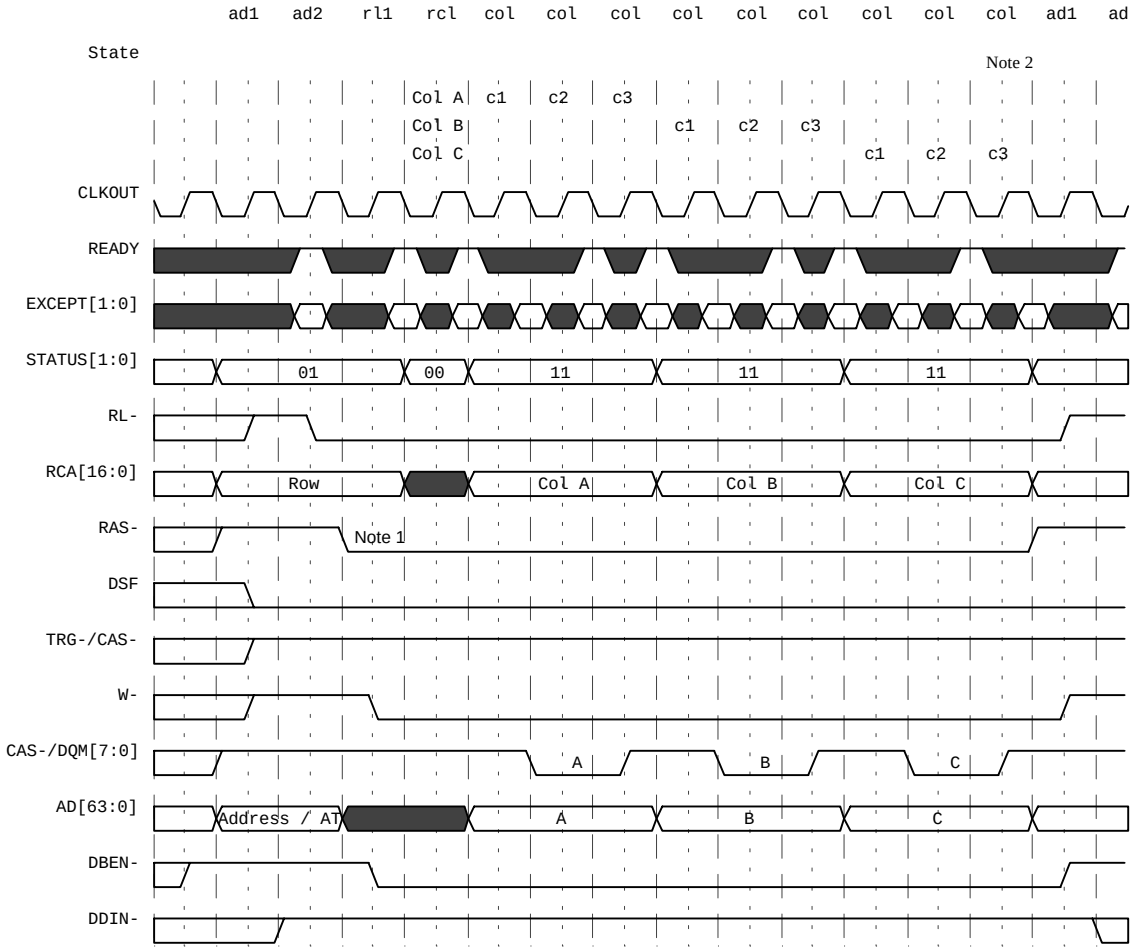
Note 1: No RAS- high time requirements are applied to these cycles.

Note 2: Additional turn-off cycles will be inserted between rto and ad1 as specified by the bank configuration.

1.2 3 cyc/col SRAM implementation: (continued)

In revision 1.x silicon, the 3 cycle/column SRAM write cycles are implemented as shown below. Note that the fall of **/CAS** is delayed by ½ cycle.

3 Cyc/Col SRAM Write - Rev 1.x



Note 1: No RAS- high time requirements are applied to these cycles.

Note 2: During peripheral data transfers, turn-off cycles will be inserted prior to ad1 as specified by the bank configuration .

Workaround:

None. For silicon revision 1.3 (see Section 1.12), a 2 cycle/column access (CT = 0110) may be used with the **WS[1:0]** field = 01 to achieve a longer **/CAS** low pulse width.

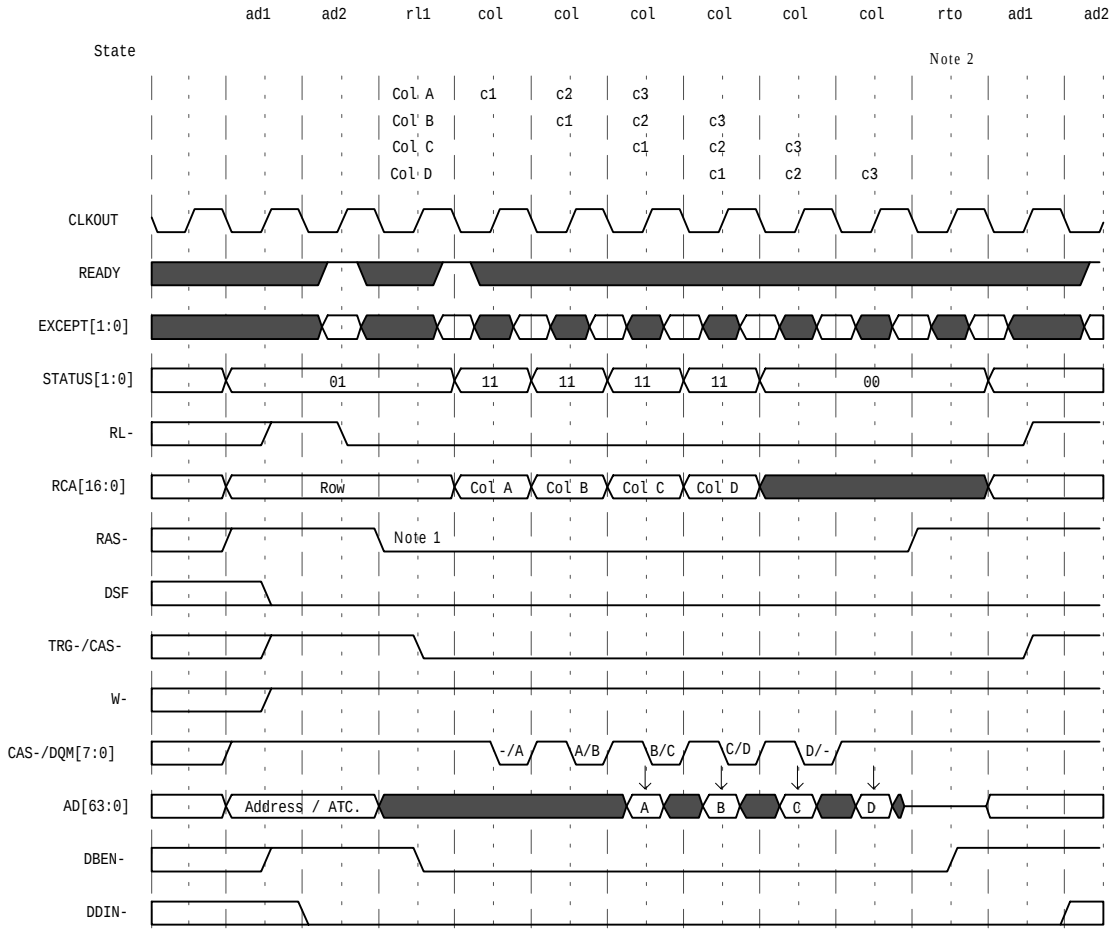
Problem exists for silicon revisions: 1.0, 1.1, 1.2, 1.3

Fix proposed : None

1.3 Synchronous SRAM implementation:

The synchronous SRAM (CT = 0100) cycles are not implemented correctly. In revision 1.0/1.1 devices, the synchronous SRAM reads are implemented as follows. Note that the ***/*CAS** signals are active (low) for only ½ cycle, and an additional ***/*CAS** pulse is inserted to force a drain that is not required.

Synchronous SRAM Read - Rev 1.0/1.1



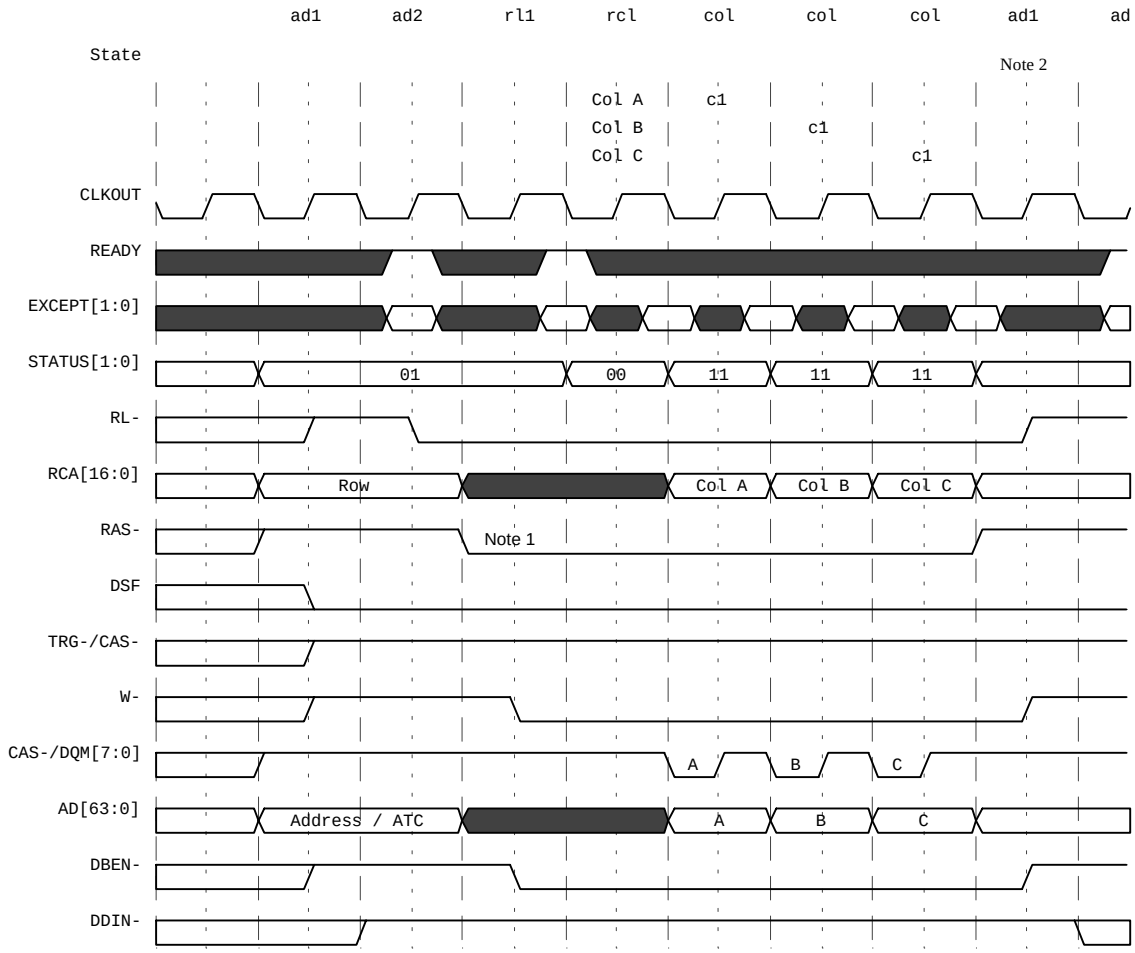
Note 1: No RAS- high time requirements are applied to these cycles.

Note 2: Additional turn-off cycles will be inserted between rto and ad1 as specified by the bank configuration.

1.3 Synchronous SRAM implementation: (continued)

In revision 1.0/1.1 silicon, the synchronous SRAM write cycles are implemented as shown below. Note that **/CAS** is active for ½ cycle, instead of remaining low for an entire machine cycle.

Synchronous SRAM Write - Rev 1.0/1.1



Note 1: No RAS- high time requirements are applicable to these cycles.

Note 2: During peripheral data transfer, turn-off cycles will be inserted prior to ad1 as specified by the bank configuration .

Workaround:
None.

Problem exists for silicon revisions: 1.0,1.1, 1.2

Fix proposed for silicon revision: 1.3

1.4 Pseudo-address corrupted after retry/fault:

A bug exists in the TMS320C82 which can force the transfer controller (TC) to output the wrong row address during a refresh cycle following a retry or fault. The corrupted address is output during the entire row access (on **RCA[16:1]**). This can **ONLY** occur for refresh cycles.

Workaround:

None. The value output on **RCA[16:1]** during refresh is supposed to be the refresh pseudo-address. Normally, several least significant (LS) bits of the pseudo-address are used for refresh bank decoding. The effect of this bug is that a bank of volatile memory may be skipped in the refresh chain depending on how the decode is performed.

The REFRATE field in the TC's REFCNTL register [0x01820000 - master processor (MP) access only] controls the trickle refresh rate of the 'C82. The value in REFRATE represents the number of machine cycles between the posting of trickle refresh requests to the TC. This value may need to be modified (decreased) to allow for overhead due to a possible missed refresh cycle due to the bank decoding of the corrupted address.

For systems that do not utilize volatile memory, or do not perform refresh bank decoding, no workaround is required.

Problem exists for silicon revisions: 1.0,1.1, 1.2

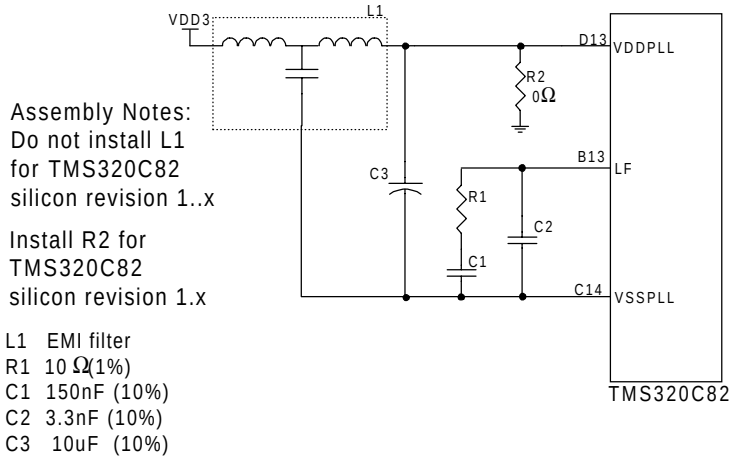
Fix verified for silicon revision: 1.3

1.5 Extensive PLL jitter:

The on-chip phase-locked loop (PLL) of the TMS320C82 exhibits a large amount of jitter in silicon revision 1.0/1.1. The jitter can prevent reliable operation.

Workaround:
Disable the PLL. This can be accomplished by connecting $\mathbf{V_{DDPLL}}$ (D13) to system GND. The following schematic illustrates this.

PLL Disabling



When $\mathbf{V_{DDPLL}}$ is tied to GND, the device operates in non-PLL mode; **CLKOUT** is ½ the frequency of **CLKIN**. With the PLL disabled, components R1, C1, C2, and C3 are not required.

Problem exists for silicon revisions: 1.0, 1.1, 1.2, 1.3

Fix proposed : None

1.6 Early transitions of DSF, /W and /TRG/CAS during EDO cycles:

The operation of silicon revision 1.x devices differs from the product datasheet description.† During extended data out (EDO) cycles, the transitions (where applicable) of **DSF**, **/W**, and **/TRG/CAS** are specified to occur midway through the rl2 state. In silicon revision 1.x, the transitions of these signals occur midway through the rl1 state.

This errata affects the following cycle types:

- 1 cyc/col pipelined EDO reads/writes
- 1 cyc/col EDO reads/writes
- 2 cyc/col EDO reads/writes
- 3 cyc/col EDO reads/writes
- all EDO block writes
- all EDO load color register (LCR) cycles

Workaround:
None required.

Problem exists for silicon revisions: 1.0, 1.1, 1.2, 1.3

Fix proposed for silicon revision: none

1.7 /RAS low time during EDO refresh cycles shortened:

The operation of silicon revision 1.x devices differs from the product datasheet description.† During EDO refresh cycles with **/EXCEPT[1:0]** = 01, the **/RAS** signal is specified to remain low for five machine cycles (through rf4). In silicon revision 1.x, the rf4 state is, in fact, not present at all, and thus **/RAS** shall remain low for only four machine cycles.

The **/RAS** high time is unaffected.

Workaround:
None.

Problem exists for silicon revisions: 1.0, 1.1, 1.2, 1.3

Fix proposed for silicon revision: none

---

† See the TMS320C82 datasheet, literature number SPRS048.

1.8 Circular buffering packet transfer support:

The operation of silicon revision 1.x devices differs from the product datasheet description.[†] For long-form packet transfers (PTs), the src/dst transfer mode (STM/DTM) field in PT Options can be set to 010 to specify a circular mode. Similarly, for short-form packet transfers, the src/dst update mode fields (SFSU/SFDU) can specify the circular mode. In silicon revision 1.x , the TC hardware to support these modes of operation has not been implemented.

Workaround:
None. Do not specify the circular addressing modes for packet transfers. For long-form PTs, the STM/DTM fields of PT Options should not be set to 010 (circular). Similarly, for short-form PTs, do not set SFSU/SFDU to codes 10 or 11 (circular).

Problem exists for silicon revisions: 1.0, 1.1, 1.2, 1.3

Fix proposed : None

1.9 Parallel processor arithmetic logic unit (PP ALU) saturation support:

The operation of silicon revision 1.x devices differs from the product datasheet description.[†] During extended arithmetic logic unit (EALU) operations, the split ALU outputs may be saturated/clamped at minimum or maximum values. In silicon revision 1.x, this mode is supported only for 32-bit signed ALU operations.

Workaround:
None. Do not specify saturation (using the **t** function modifier) for any modes other than the 32-bit signed case.

Problem exists for silicon revisions: 1.0, 1.1, 1.2, 1.3

Fix proposed : None

[†] See the TMS320C82 datasheet, literature number SPRS048.

1.10 Incorrect packet transfer parameter reload:

A bug exists in the TMS320C82 wherein a packet transfer which has been suspended may be restarted from the beginning rather than from the suspension point. This errata is the result of a small timing window in which the following occurs:

1) The PT suspends; the linked-list pointer in the requesting processor's parameter RAM is modified to point to its suspend area.
2) The resubmission process begins, and the TC modifies the requesting processor's linked-list pointer to point to the original parameters (in the requesting processor's parameter RAM).
3) As the resubmission process continues, a higher priority PT [or externally initiated packet transfer (XPT)] comes in, which forces the TC to abandon the resubmission, and begin servicing the higher priority transfer. The linked-list pointer is *NOT* modified as it should be to point back to the suspend area.
4) Upon returning to the suspended transfer, the linked-list pointer is evaluated and the *ORIGINAL* parameters are read into the TC registers instead of the parameters from the suspend area.

The net effect of the above sequence of events is to cause the suspended PT to restart from the beginning. For systems that utilize FIFOs, this can cause the system to fail as extra data may be written to/requested from the FIFO when the PT restarts. For non-FIFO-based systems, the only side effect of this errata is that it may incur extra latency on completion of the suspended PT.

Workaround:

Ensure that PTs cannot suspend, by adjusting PTMIN to a value large enough to allow for completion. In some systems, this may result in an unusually large value for PTMIN; for these systems, it may further be necessary to break up large PTs into several smaller PTs and then apply the same workaround.

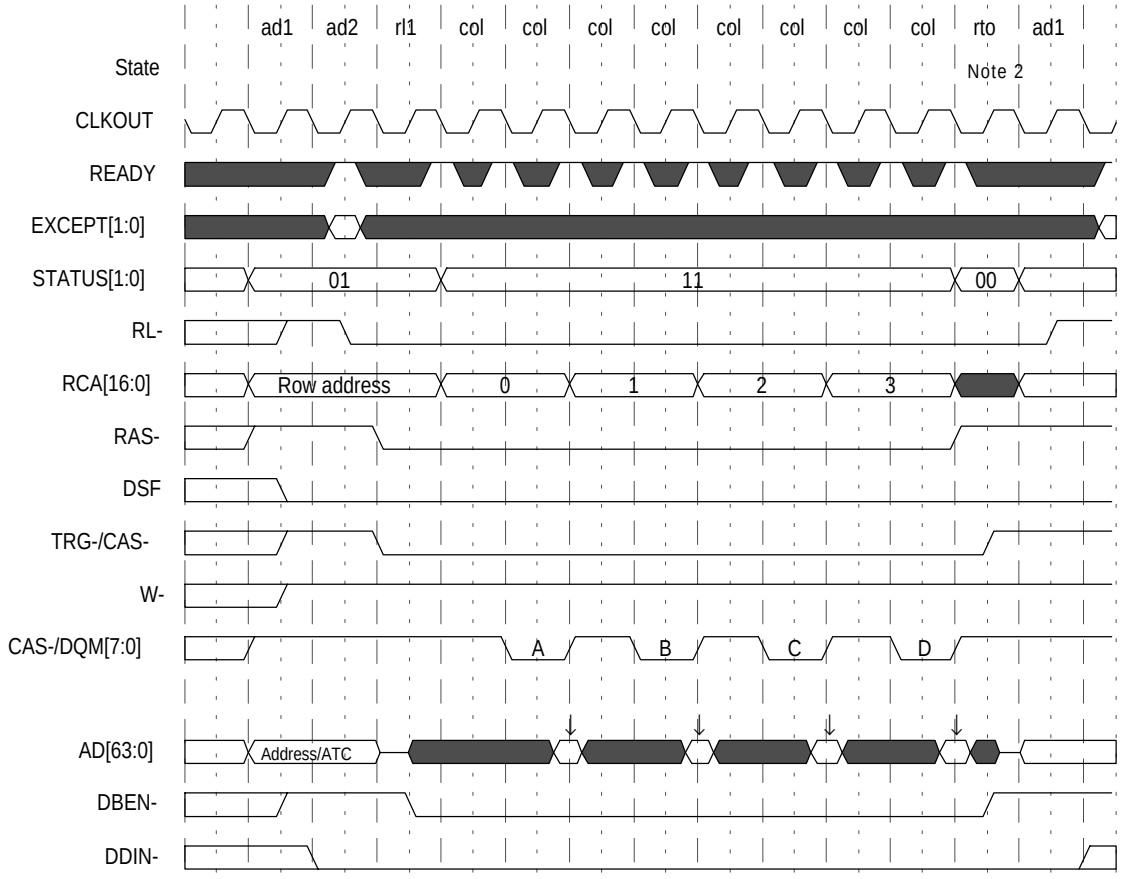
Problem exists for silicon revisions: 1.0, 1.1, 1.2

Fix verified for silicon revision: 1.3

1.11 Memory bank configuration cycle:

The memory bank configuration cycles are not implemented correctly. Similar to the 2 cycle/column SRAM reads, the fall of **/CAS** is delayed by ½ cycle. The following cycle diagram illustrates the operation of revision 1.0/1.1 silicon for memory bank configuration cycles.

Bank Configuration Cycle - Rev 1.0/1.1



Note 2: Additional turn-off cycles can be inserted by adding waitstates during the rto (turnoff) cycle. This capability is unique to this cycle.

Note: This diagram represents the errata timing for Rev 1.0 & Rev 1.1 Devices. For Rev 1.2 devices, the timing will appear as shown in the TMS320C82 datasheet.

Workaround:

None. Devices with faster access times may be needed to provide configuration data if **/CAS** is used as the output enable of said device.

Problem exists for silicon revisions: 1.0, 1.1, 1.2

Fix verified for silicon revision: 1.3

1.12 Programmable wait states not implemented:

The operation of silicon revision 1.0/1.1 devices differs from the product datasheet description.† The memory bank configuration cache bits for the **WS[1:0]** (wait state) and R bit (row wait) are not implemented in revisions 1.0/1.1 of the TMS320C82, nor is the functionality they provide.

Workaround:
None. The **WS[1:0]** and R bit positions in the configuration cache data should be set to 0s (zeroes) to ensure proper operation in revisions 1.0/1.1 of the TMS320C82.

Problem exists for silicon revisions: 1.0, 1.1, 1.2

Fix verified for silicon revision: 1.3

---

† See the TMS320C82 datasheet, literature number SPRS048.

1.13 Device hangs after reset:

The following bug has been found in revisions 1.0/1.1 of the TMS320C82 wherein the device may hang after reset.

The TMS320C82 samples the **/HREQ** (host request) input on each rising edge of **CLKOUT**. On the rising edge of **CLKOUT** immediately preceding the rise of **/RESET**, the **/HREQ** input is used to determine the power-up state of the MP. If **/HREQ** is sampled low, the MP will power up in a running state, and will perform its boot instruction fetch from 0xFFFFFFC0 (subblock address containing 0xFFFFFFF8) immediately after completing the default refresh initialization cycles. If **/HREQ** is sampled high at this time, the MP will remain in a halted state until the first rising edge on **/EINT3**. The refresh initialization cycles will still occur, as will trickle refreshes thereafter.

The hold time of the **/HREQ** input with respect to the rising edge of **/RESET** is 0 ns, such that the TMS320C82 may be powered up running by tying **/HREQ** directly to **/RESET**. A bug has been found in the sampling logic of **/HREQ** which will cause the device to hang if the rise of **/HREQ** is sampled low on the second **CLKOUT** cycle after the rise of **/RESET** before the refresh initialization cycles. This is illustrated below.

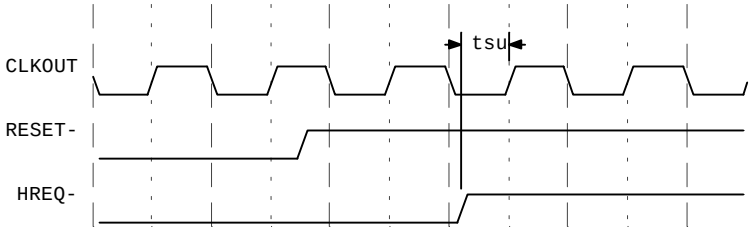


Figure 1.13.1 - Normal Operation

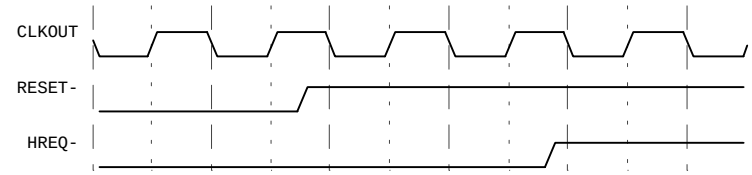


Figure 1.13.2 - Device Hangs After Reset

When the second case is encountered, the TMS320C82 will begin driving all signals normally in the high-impedance state once the host request condition is removed (**/HREQ** pulled high). The device will begin the first refresh cycle normally performed after reset but will then hang. All outputs of the TMS320C82 will continue to be driven, though no signal transitions other than **CLKOUT** will occur.

Workaround:

Ensure that **/HREQ** is sampled high on the second **CLKOUT** rising edge after the rise of **/RESET**. Note that tying **/HREQ** directly to **/RESET** is an acceptable workaround. For multi-C82 designs, it may be required that separate **/RESET** signals be provided for each TMS320C82 to prevent the refresh initialization cycles from conflicting with one another.

Problem exists for silicon revisions: 1.0, 1.1, 1.2

Fix verified for silicon revision: 1.3

1.14 Default turn-off cycles in read cycles:

The operation of silicon revision 1.x devices differs from the product datasheet description.† During all read cycles in Rev 1.x, one default turn-off cycle is always inserted at the end of a page of accesses. This occurs regardless of the state of the TO(1:0) field in the memory bank configuration cache entry for the addressed bank. The TO(1:0) field correctly inserts turn-off cycles *in addition* to the single default turn-off cycle which is always inserted on read cycles.

Workaround:
None.

Problem exists for silicon revisions: 1.0, 1.1, 1.2, 1.3

Fix proposed for silicon revision: none

1.15 /TRG active during turn-off cycles:

A bug has been found in the Rev. 1.x silicon. At the end of a page of read accesses, turn-off cycles may be added using the TO(1:0) field in the bank configuration cache entry for the addressed bank of memory (by default, in Rev 1.x one turn-off cycle will always be present during read cycles; see Section 1.14). This feature allows time for the **AD[63:0]** bus to turn off before being driven by the 'C82 during the subsequent row access. During turn-off cycles, the **/TRG** signal should be high, to disable output drivers for memories which require/use this signal as an output enable.

A bug exists in revisions 1.0/1.1 wherein the **/TRG** signal is *NOT* driven inactive high during turn-off cycles. This is true for both the default turn-off cycle inserted by the 'C82 during read cycles (see Section 1.14) and any additional turn-off cycles specified in the TO(1:0) field of the bank configuration entry for the addressed bank.

Workaround:
The **/TRG** is most often used with industry standard VRAMs as a select between normal cycles and transfer cycles [shift register transfers (SRTs)]. Additionally, it serves as an output enable for the VRAM. In some systems, it is used as an output enable for other memory types.

During turn-off cycles, **/RAS** and **/DBEN** are driven high, and thus for VRAM interfaces, no problem exists as **/RAS** will disable outputs from the RAM. For other memories and peripherals utilizing the **/TRG** output, a logical combination with the **/RAS** or possibly **/DBEN** may be used to produce an output-enable signal suitable to ensure that no drive conflict will occur on the **AD[63:0]** bus.

Problem exists for silicon revisions: 1.0, 1.1, 1.2, 1.3

Fix proposed for silicon revision: none

---

† See the TMS320C82 datasheet, literature number SPRS048.

1.16 Timing window during XPTs causes bus idle condition:

The following bug exists in Rev 1.0/1.1 silicon. During the completion of an externally initiated packet transfer (XPT), a one-cycle timing window has been found during which an incoming cache or direct external access (DEA) request may be temporarily ignored. Cache/DEA requests submitted during any cycles outside of this window are not affected. Refresh cycles are not affected.

The most noticeable ramification of this bug is the delayed servicing of the cache/DEA request. If a cache/DEA request comes in during this one-cycle window, it will be temporarily ignored by the TC. The active XPT will complete normally; however, the cache/DEA request may not be serviced until another cache/DEA request, or another XPT request comes into the TC. In the former case, the second cache/DEA request will be prioritized with the stalled cache/DEA request accordingly; in the latter case, the XPT will actually be serviced first because it has priority over the cache/DEA request(s).

Workaround:
Submit a second cache/DEA request to “unstall” the TC. Since a cache/DEA request stalls the requesting processor, this workaround requires the use of a second processor (typically one of the PPs) to periodically submit a dummy DEA which will force the TC to reprioritize all pending inputs, potentially including the stalled cache/DEA request which came in on the aforementioned one-cycle window.

For systems that do not use XPTs, no workaround is required.

Problem exists for silicon revisions: 1.0, 1.1, 1.2

Fix verified for silicon revision: 1.3

1.17 V_{OH} degradation on output pins following 0-1 transition

Under certain conditions associated with output drivers switching from a logic zero to a logic one, the V_{OH} level of said outputs may not meet the V_{OH} specification of 2.4 V.

The existence of this errata is the result of transient negative-going glitches observed on the output pins following a 0-1 transition. The steady state value of the output is unaffected, and specified to be a minimum of 2.4 V.

Figure 106, titled 'TTL Level Outputs' in the TMS320C82 datasheet (literature number SPRS048), shows 2.4 V as the minimum V_{OH} level used to ensure output pin timing. Instead of testing to 2.4 V, the following table describes the specified values (tested in manufacturing test):

V_{DD}	V_{OH} (AD[63:0], RCA[16:0])	V_{OH}, All other outputs
V_{DD} < 3.3 V V_{DD} = 3.3 V V_{DD} > 3.3 V	1.8 V	1.9 V

Workaround:
None.

Problem exists for silicon revisions: 1.0, 1.1, 1.2

Fix proposed for silicon revision: 1.3

1.18 V_{OL} degradation on output pins following 1-0 transition

Under certain conditions associated with output drivers switching from a logic one to a logic zero, the V_{OL} level of said outputs may not meet the V_{OH} specification of 0.6 V. A value of 0.8 V is specified instead (tested in manufacturing test).

The existence of this errata is the result of transient negative-going glitches observed on the output pins following a 1-0 transition. The steady state value of the output is unaffected, and specified to be a maximum of 0.6 V.

Workaround:
None.

Problem exists for silicon revisions: 1.0, 1.1, 1.2

Fix proposed for silicon revision: 1.3

1.19 Timing window during PT parameter load causes prioritization logic to ignore XPTs: UPDATED!

The following bug exists in Rev 1.x silicon. During the load cycles of an MP- or PP-initiated packet transfer, a one-cycle timing window exists during which an incoming XPT request will cause the processor packet transfer to run to completion regardless of the setting in PTMIN. This has the effect of circumventing the TC prioritization logic, which can cause undesirable system effects by delaying the servicing of higher priority packet transfers, including XPTs.

The errata exists due to a timing window that occurs on the cycle in which PTMIN is copied into an internal register. The internal register is used by the TC to determine points at which re-prioritization of events may occur. If a new XPT is synchronized on this cycle, the loading of the internal register with PTMIN is corrupted. As a result, the packet transfer whose parameters are loaded into the TC registers will run for a long period of time. (In most cases, the transfer will run to completion as the maximum value for PTMIN translates to about .33 seconds, which is greater than the duration of most packet transfers.) Thus, any higher priority packet transfers from another processor, or the **/XPT[3:0]** inputs, will be ignored until the active packet transfer is completed (or 2^{24} cycles elapse), at which point the TC will correctly reprioritize any pending events.

Workaround:

The workaround method is to utilize the TC's PTMAX register to force periodic suspension of the MP/PP-initiated packet transfers. The TC will suspend an MP- or PP-initiated packet transfer (XPTs cannot be suspended) after PTMIN + PTMAX cycles of service for said transfer. If no other accesses are pending, then the PT is automatically resubmitted from the point at which it was suspended. If other higher priority events are waiting to be serviced, however, they will be serviced prior to the automatic resubmission of the PT. In this manner, this bug can be worked around by forcing the TC to suspend an active PT and reprioritize all pending requests.

To apply this workaround, the system designer should ensure that PTMIN + PTMAX is less than the maximum number of cycles it is acceptable to delay XPT service by. This will vary from system to system.

For systems that do not use XPTs, no workaround is required.

Problem exists for silicon revisions: 1.0, 1.1, 1.2, 1.3

Fix proposed for silicon revision: None

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current and complete.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.