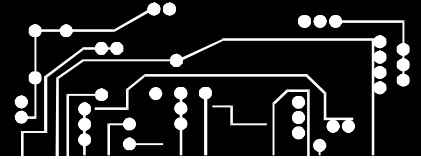


PowerPC Embedded Processors

Application Note



DRAM Controller for the PowerPC 401GF

IBM Microelectronics
Dept D95/Bldg 060
3039 Cornwallis Road
Research Triangle Park, NC 27709
Version: 1.0

November 18, 1996

Today's embedded systems require low-cost, low-power, high-performance controllers to meet market demands. The IBM PPC401GF 32-bit RISC embedded controller meets these critical requirements for many embedded system designs. This application note describes a DRAM controller designed to interface directly to the 401GF. Along with the 401GF, this DRAM controller provides a solid foundation upon which to build prototypes, and thus reduce development time.*

*The design was implemented using state machines described in Device Synthesis Language (DSL), a high-level device language developed by MINC**. Most integrated CAE tools, such as those from Cadence, Mentor Graphics and Viewlogic, incorporate a MINC fitter engine making programmable logic designs transportable. DSL enables end users to easily incorporate this DRAM controller design into their existing CAE tool design, using the MINC fitter engine to program their PLD of choice.*

*The DSL equations were developed using the AMD** MACHXL** 3.0 software. Functional simulation was done using the MACHXL package and timing verification was accomplished using Chronology's TimingDesigner**.*

The DRAM controller design was compiled and fitted into a MACH210-12 PLD. Together with one DRAM component, this small, low-cost part creates a memory subsystem that complements the 401GF well. End users can obtain the source level equations and program their own MACH210 device directly or make modifications and program a PLD that more directly meets their needs using their existing CAE tool. Designs are brought to market faster because design time is not spent purchasing and learning a new programmable logic tool.

1. Design Goals

This DRAM controller design was developed to meet the following goals:

- Use standard components to minimize prototype delay
- Provide basic DRAM controller equations that can be easily incorporated into larger designs to minimize customer's ASIC or FPGA design cycle time
- Provide DRAM controller equations along with a suggested PLD manufacturer and part number to allow end users to immediately begin prototyping with the 401GF
- Demonstrate the advantages of using the 401GF in a high-performance, low-cost design

The DRAM controller design incorporates the following features:

- Burst read timing of 6-2-2-2
- Burst write timing of 5-2-2-2
- Interfaces to standard 70ns EDO DRAM
- Provides control for one bank of non-interleaved memory
- All signals generated from a single 33MHz oscillator
- 33MHz bus speed

2. Theory of Operation

Figure 1 shows a typical system design using the 401GF and this DRAM controller design. External latches latch the address during the address phase of the transaction using the /ALE signal (where / indicates an active low signal) supplied by the 401GF. The DRAM logic controls external multiplexers to direct the row and column addresses to the DRAM components. Data from the DRAM returns to the 401GF on the address/data bus. Likewise, data written to DRAM is sent across the multiplexed bus during the data phase of the transaction and latched by the CAS signals supplied by the DRAM controller.

The following subsections describe each component in further detail.

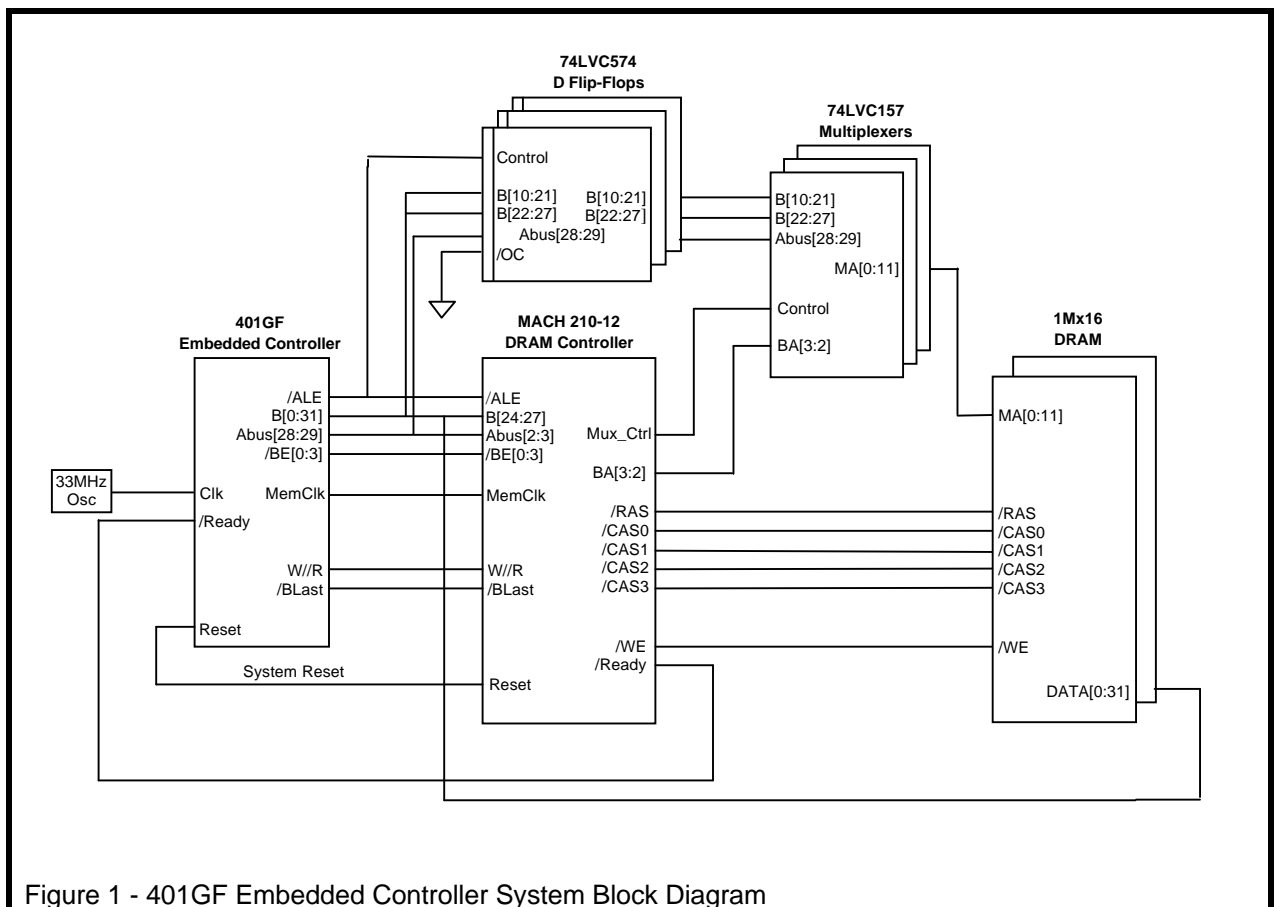
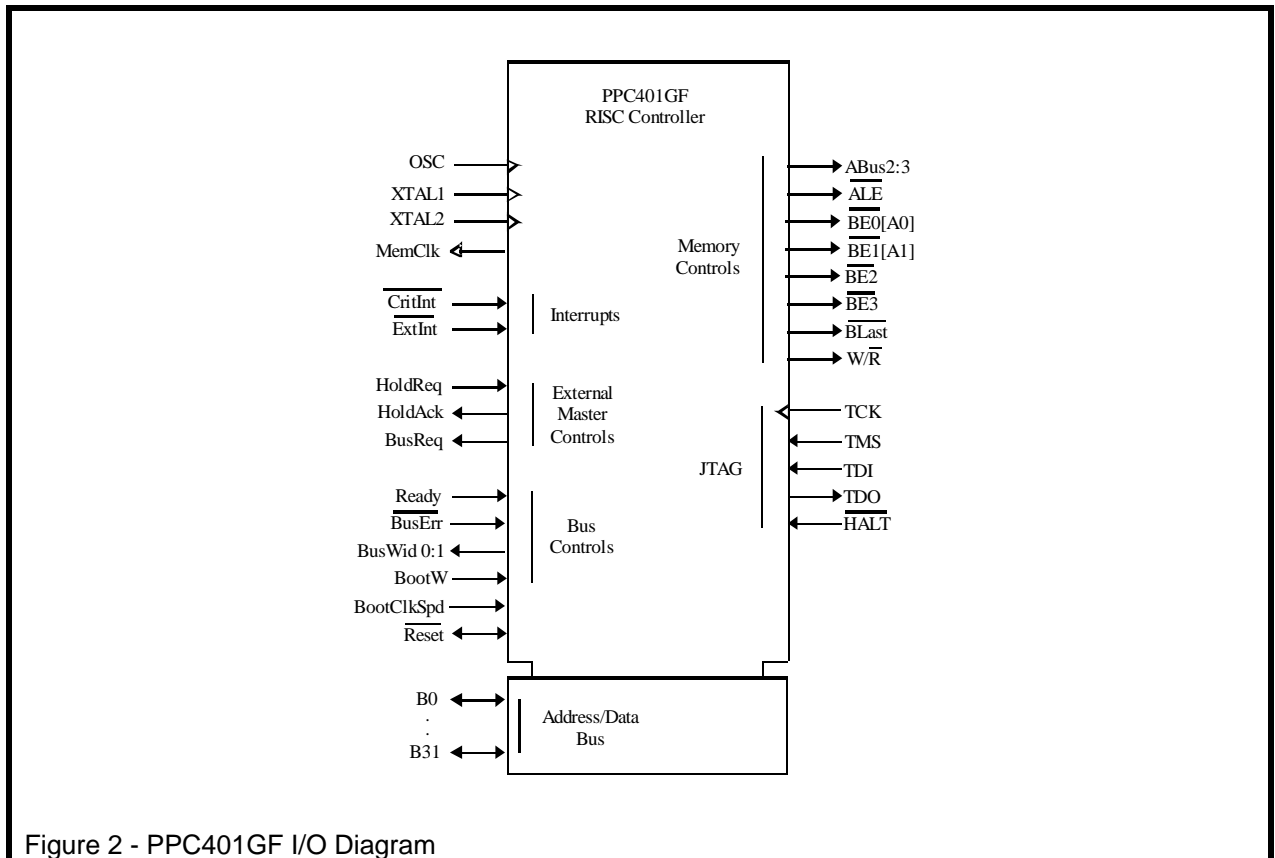


Figure 1 - 401GF Embedded Controller System Block Diagram

2.1 PPC401GF Embedded Controller

The 401GF embedded controller supports a 32-bit, device-paced, multiplexed address and data bus capable of bursting as many as four words consecutively when implemented in a 32-bit system. For this application, an external bus speed of 33MHz was assumed, although the 401GF can support slower speeds.

To initiate a transaction, the 401GF activates /ALE and drives the address and control signals on a rising clock edge. During byte, halfword, 3-byte and word writes, the 401GF drives valid data on the next clock edge and holds data until it senses an active /Ready signal from the DRAM controller on the rise of a clock. During a burst write, the 401GF drives the next piece of data on the same clock edge it senses an active /Ready. It ends the transaction when it senses an active /Ready while in the process of driving /BLast active. The write timing diagrams in Figure 7 and Figure 9 illustrate the timings.



During byte, halfword, 3-byte and word reads, the 401GF initiates a transaction similar to writes, but activates /BLast immediately and holds it low until it samples an active /Ready on a rising clock edge. For burst reads of n words, where $n = 2, 3, \text{ or } 4$, the 401GF drops /BLast on the same clock edge as it clocks in the $n-1$ piece of data, and raises /BLast on the clock edge that it samples an active /Ready. Figure 6 and Figure 8 illustrate the timings.

2.2 MACH210-12 DRAM Controller

The DRAM controller attaches directly to the 401GF minimizing the number of external components. Two state machines control the necessary DRAM accesses required by the 401GF while providing the CAS-before-RAS refresh function for the DRAM modules. The DRAM controller also provides the initialization delays and synchronization refreshes required by the DRAM components during initial power-up. A detailed discussion of the DRAM controller is found in Section 3.

The DRAM controller supports the following 401GF-initiated transactions:

- Byte, halfword, 3-byte and word reads
- Byte, halfword, 3-byte and word writes
- Burst reads of up to 4 words
- Burst writes of up to 4 words

2.3 74LVC574 D flip-flops

The 74LVC574 Octal D-type flip-flops are 3-state, 8-bit, positive edge-triggered flip-flops that latch the address during the address phase of a transaction. Three external components are required to latch the output signals used to produce the row and column addresses. The /OE pins on the latches are tied to ground allowing the latched address to remain valid at the output of the flip-flops. The rising edge of the 401GF's /ALE is used as the clock input to the flip-flops to latch the data.

2.4 74LVC157 Multiplexers

The 74LVC157 quad two-input multiplexers multiplex the address from the D flip-flops into row and column addresses. Each component multiplexes four bits, so three components are needed to multiplex the 12 address bits required by the DRAM components. The multiplexers are controlled by an output signal from the DRAM controller which switches the row address to the column address at the appropriate times.

2.5 1Mx16 70ns EDO DRAM Components

Two 1Mx16 70ns EDO DRAM components are arranged in one bank to provide x32 access to two MBytes of main memory. These particular components are asymmetric, requiring twelve row address lines and eight column address lines. The DRAM controller provides one RAS signal, four CAS signals, and the appropriate CAS-before-RAS (CBR) refresh signals.

2.6 33MHz Oscillator

A 33Mhz oscillator which clocks the 401GF and can run in 1:1 mode (the CPU runs at the same frequency as the input clock), or 2:1 mode, where the 401GF runs twice as fast (66MHz.) Note that the DRAM controller logic is set to run at a maximum bus speed of 33MHz. If the system designer chooses to use the MemClk output of the 401GF to drive the DRAM controller PLD, the user must insure that MemClk must not exceed 33MHz by setting the correct register values in the 401GF.

3. DRAM Controller State Machines in Detail

The DRAM controller design consists of two state machines:

- Refresh Counter State Machine (see Figure 3)
- Main State Machine (see Figure 5)

The *Refresh Counter State Machine (RCSM)* controls the required initialization and refresh delays by maintaining the refresh counter.

The *Main State Machine (MSM)* controls the required signals to the DRAM components for both 401GF initiated accesses and DRAM refresh accesses.

At power-up, this controller design satisfies the DRAM initialization requirements by providing more than a 200 μ s delay followed by 9 CAS-before-RAS refreshes (DRAM requires a minimum of 8) before allowing the 401GF access to DRAM. The RCSM controls a counter along and an enable signal that enables 401GF-initiated accesses only after meeting both initialization criteria.

After initialization, the MSM remains in the **IDLE** state waiting to perform either a refresh or a 401GF initiated DRAM access. If neither the 401GF is requesting an access or the refresh timer is not expired the MSM remains in the **IDLE** state. A refresh request has priority over a DRAM access, which implies that if the refresh timer expires at the same time as a DRAM access request is sensed from the 401GF, the controller proceeds with the DRAM refresh, holding off the processor until the refresh completes.

The 401GF initiates an external transaction by lowering /ALE, driving a valid address on the external bus (B[0:31]), and activating the control signals at the rising clock edge. Because /ALE is deactivated on the next falling edge, address data strobe (/ADS) is created internally to the DRAM controller. /ADS senses the high to low transition of /ALE and remains low until the next rising clock edge. The DRAM controller detects the start of a 401GF transaction when /ADS is active at the rise of the clock.

To insure proper timing and minimize pincount on the MACH device, the DRAM controller only decodes the four most significant bits of the address (B[24:27]). This results in a DRAM region of 256MBytes set at 0x0xxxxxxx.

3.1 Refresh Counter State Machine

The main function of the Refresh Counter State Machine, shown in Figure 3, is to control delay counters for the MSM. During initial power-up, the DRAM components require a 200 μ s delay followed by a minimum of eight CBR refreshes before allowing any DRAM access by the 401GF. This 1-bit state machine provides the necessary delays to insure that the DRAM components are initialized properly along with maintaining the refresh counter.

The RCSM consists of two states; the **ref_delay** state and the **ref_acc** state. The state machine remains in the **ref_delay** state while the refresh counter (Count_A) is counting. When the counter expires, the RCSM transitions to the **ref_acc** state informing the MSM that a refresh operation needs to be performed. After the MSM performs the refresh operation, the RCSM resets Count_A, transitions back to the **ref_delay** state, and begins counting again.

Count_A is a 9-bit counter that expires when it reaches 512 (15.6 μ s at the 30ns clock period.) Therefore, during normal operation the RCSM requests a refresh operation approximately every 15.6 μ s.

Count_B is a 5-bit counter that controls the MSM during power-up. The MSM monitors Count_B and alters its mode of operation depending on Count_B's present value.

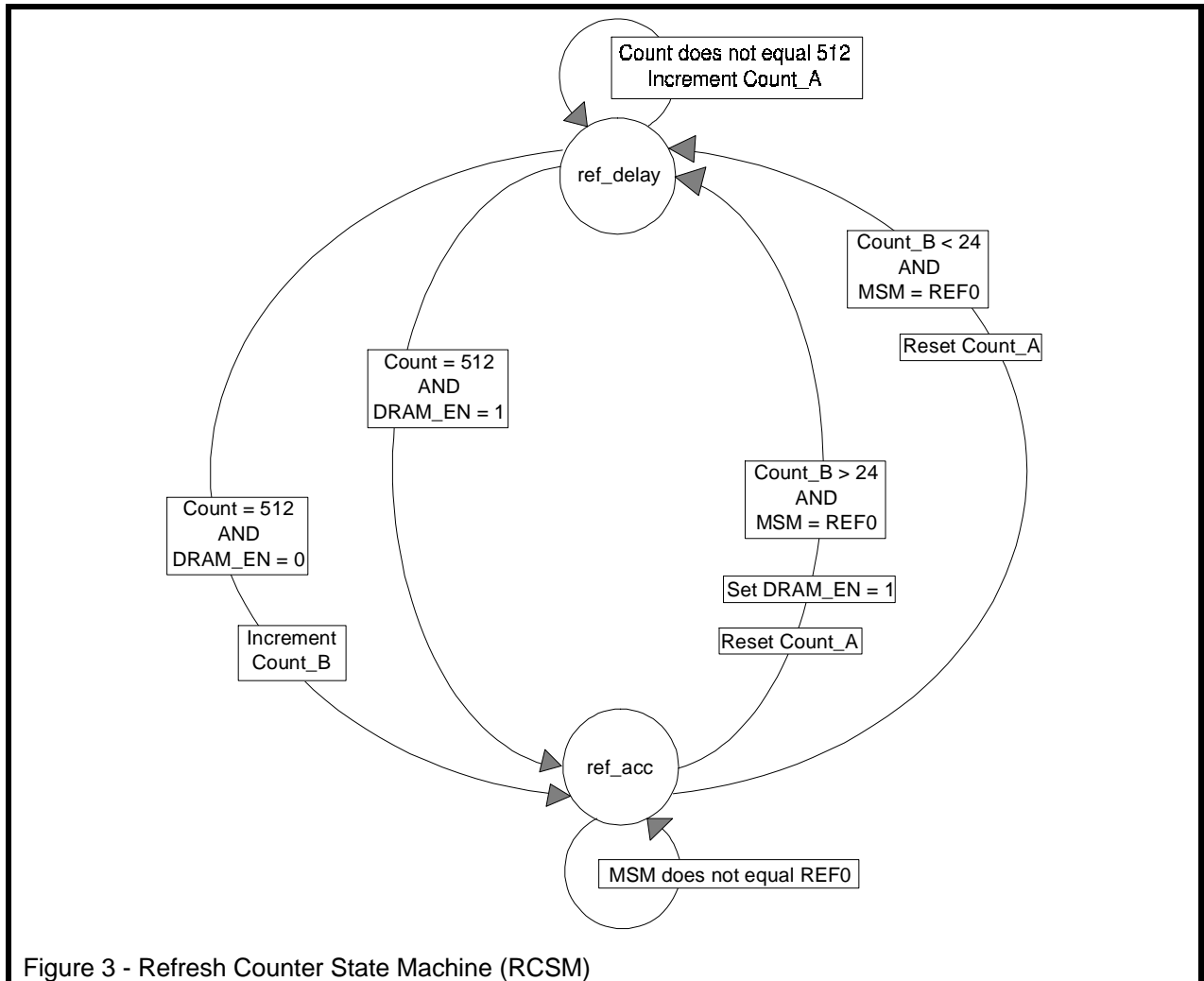
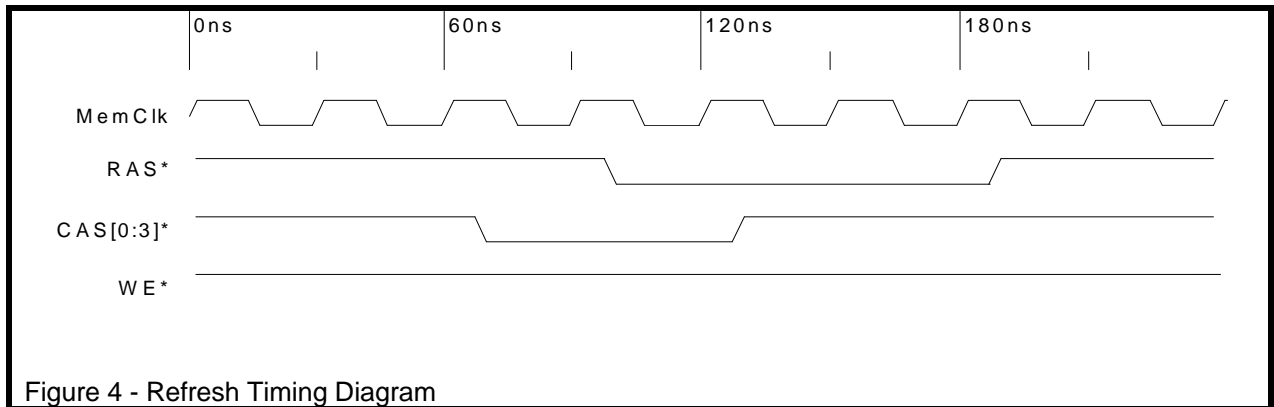


Figure 3 - Refresh Counter State Machine (RCSM)

Initially, Count_B is set to 0. When Count_A expires (after 512 clock cycles), Count_B is incremented while the RCSM transitions to the **ref_acc** state. The MSM ignores the first 16 refresh requests from the RCSM (while Count_B < 16) by immediately forcing the RCSM to transition back to its **ref_delay** state without performing a DRAM refresh. Forcing the RCSM to run through its 512-cycle delay 16 times causes an approximate delay of 245μs, which is greater than the 200μs delay that the DRAM components require.

For the next eight counts of Count_B (16 < Count_B < 24), the MSM performs CBR refreshes while the RCSM is in the **ref_acc** state. The RCSM still operates in the same manner (when Count_B < 16) but monitors Count_B itself, waiting for Count_B = 24. At this point, the MSM will have performed more than eight CBR refreshes. The RCSM sets a DRAM_EN signal active on a transition from **ref_acc** back to **ref_delay** after nine CBR refreshes are performed (Count_B > 24). The MSM monitors the DRAM_EN signal in its **IDLE** state, and performs DRAM accesses for the 401GF when the MSM sees DRAM_EN active. After the activation of DRAM_EN, the RCSM begins normal refresh operation and Count_B no longer alters the normal operation of the MSM. The RCSM no longer increments Count_B.

If the 401GF finishes its initial power-up routine and attempts to perform a memory access while the DRAM controller is in the process of initializing DRAM, the controller holds off the 401GF by not supplying a /Ready signal. The DRAM controller notes the DRAM access pending and immediately performs the access for the 401GF when the DRAM controllers initialization routine completes.



3.2 Main State Machine: DRAM Access

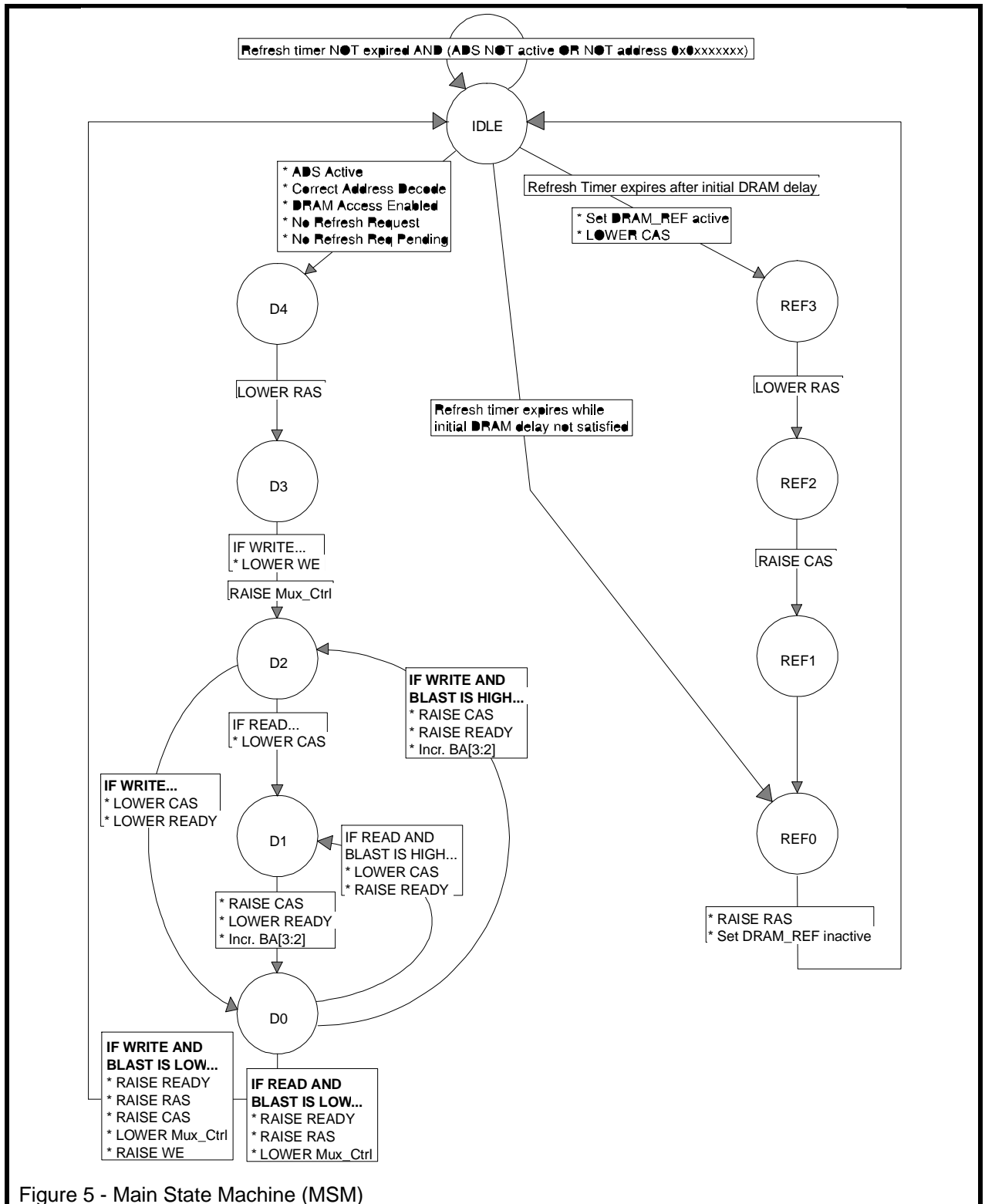
If a valid DRAM address is decoded, an active /ADS is sensed, and a refresh request is not sensed, the MSM leaves the **IDLE** state and proceeds with the DRAM access to state **D4** providing that a 401GF access is enabled. By this time the address has been latched into the D flip-flops with the rising edge of the /ALE signal. Because Mux_ctrl is low, the proper row address makes its way through the flip-flops (/OC tied to ground), through the external multiplexers, and presents itself to the DRAM input. Because propagation delay through the external modules does not allow /RAS to be driven active leaving the **IDLE** state, the earliest /RAS can be driven active is during the **D4** to **D3** transition. The Mux_ctrl signal switches at the rising edge of the **D3** to **D2** state transition presenting the column address to the DRAM.

The type of transaction taking place on the bus determines the next state transition. If the DRAM access is a write (W//R = 1), the MSM jumps to state **D0**, skipping state **D1**. This one-cycle savings is contributed to the 401GF's ability to present the data immediately following the address cycle for a write, as opposed to when the 401GF has to wait for DRAM to access its data on reads. The transition from **D2** to **D0** during the write cycle brings /CAS active, causing the DRAM module to latch in the data. The DRAM controller also makes /Ready active on the **D2/D1** clock edge, which informs the 401GF on the next rising clock that the DRAM controller is ready for the next word. If the 401GF is in the middle of a burst, the DRAM controller senses /BLast high and returns to state **D2**. In the process, it raises /CAS and /Ready back to their inactive states and supplies the next word address to the DRAM through the external multiplexer by incrementing its BA[3:2] outputs. If the 401GF is performing a byte, halfword, 3-byte, or word read, the DRAM controller senses /BLast low and brings /RAS, /CAS, Mux_ctrl, and /Ready back to their inactive states and proceeds immediately to the **IDLE** state.

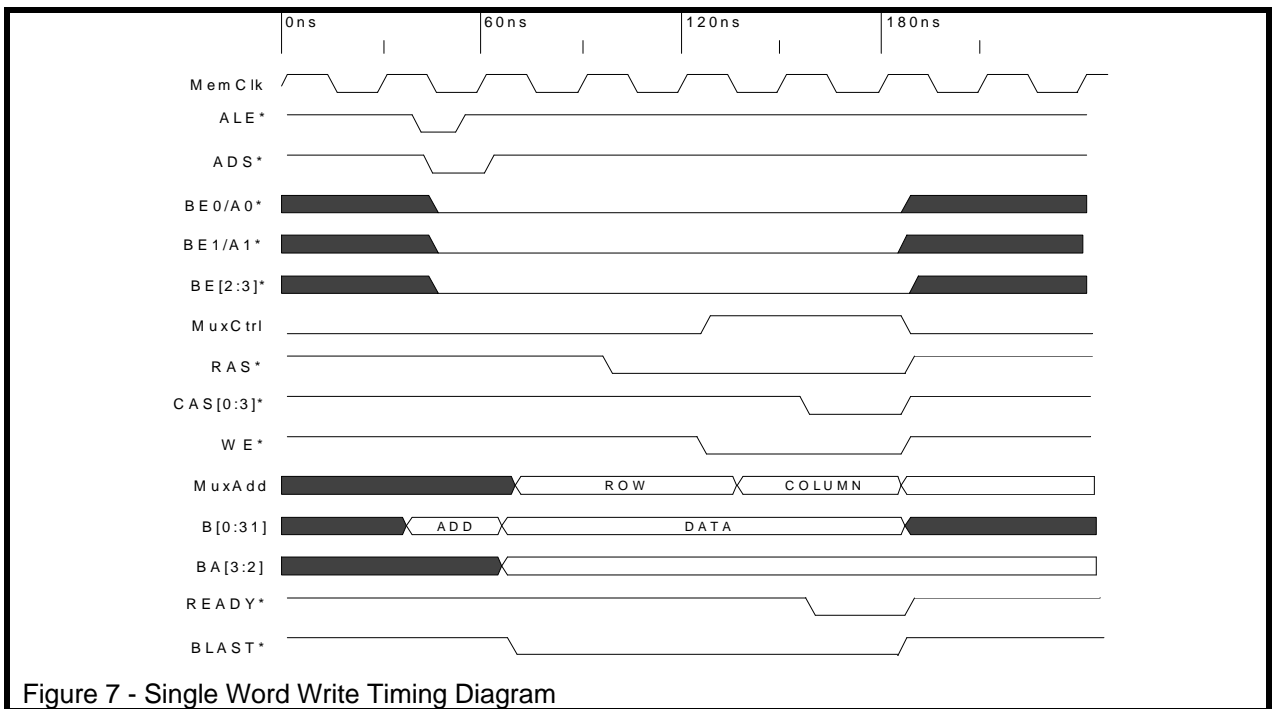
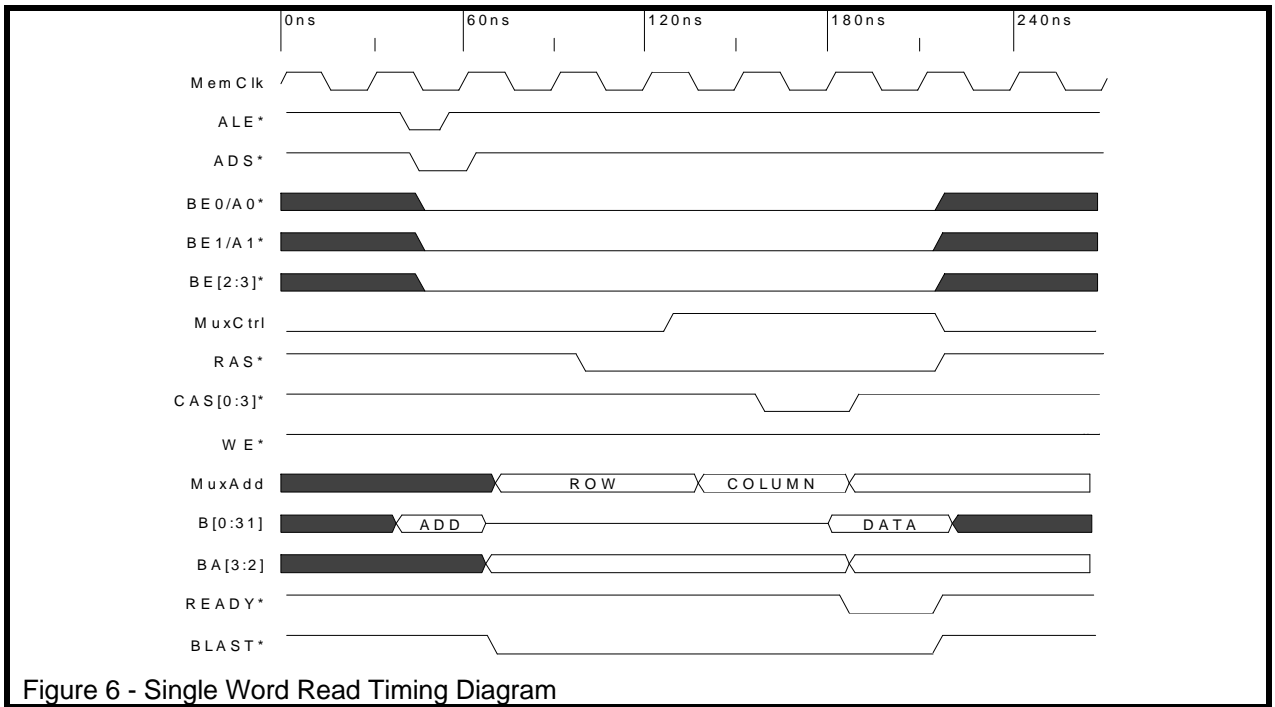
If the 401GF is performing a read (W//R = 0), the DRAM controller leaves state **D2** and enters state **D1** while bringing the necessary /CAS signals active. During the next state transition, **D1/D0**, the controller will raise /CAS and lower its /Ready signal, informing the 401GF that the data is now available from the DRAM to be read. For burst reads, the DRAM controller increments the word address bits BA[3:2] at the same clock edge. During non-burst reads, this incremented address has no affect since the transaction will end on the next cycle.

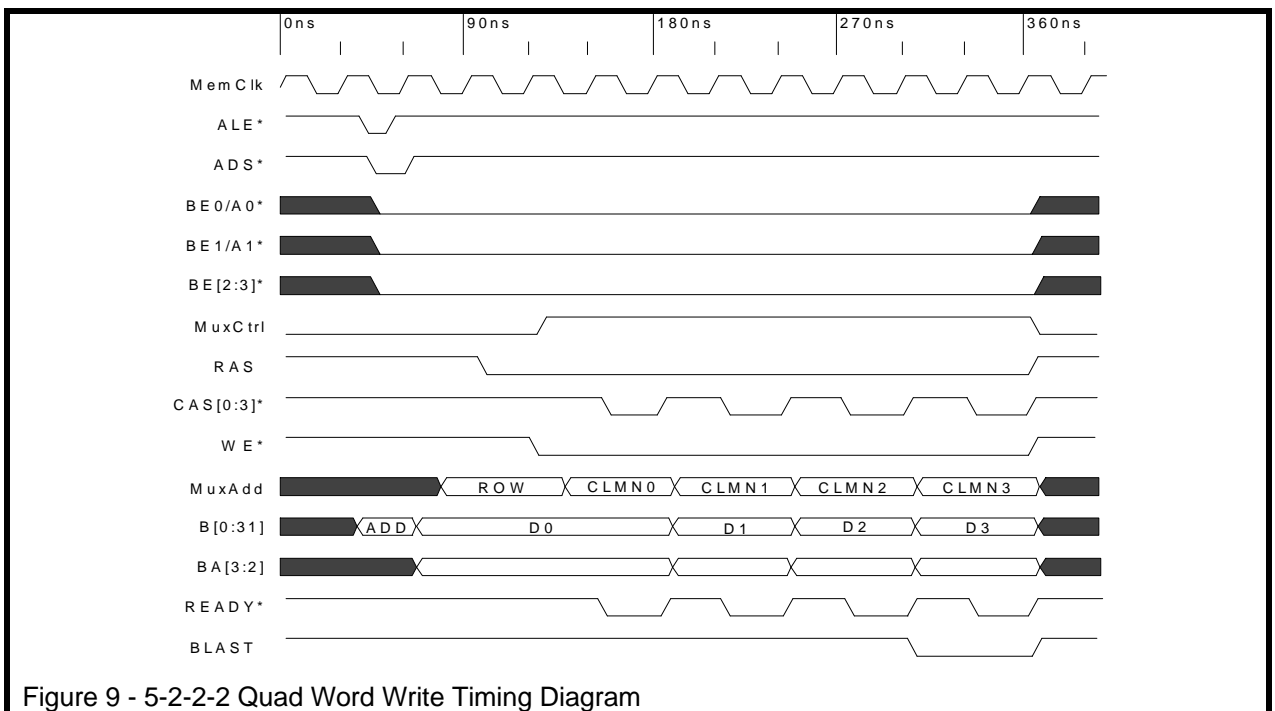
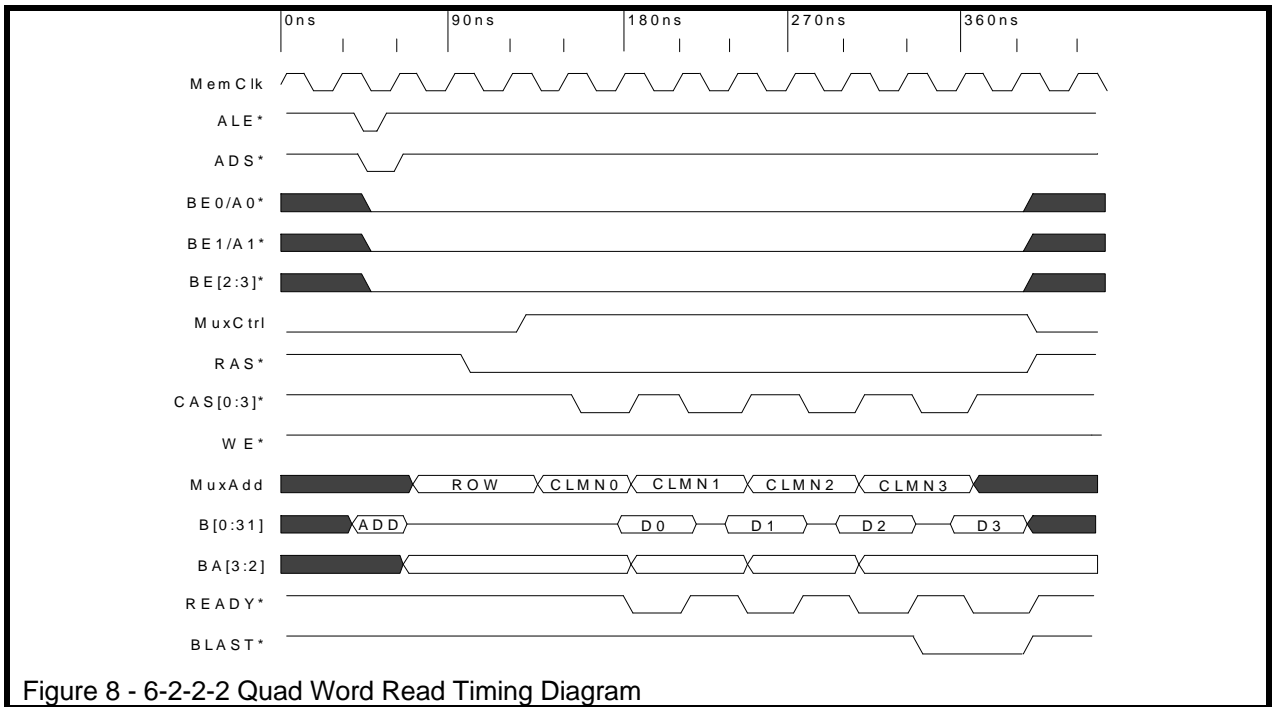
In state **D0**, the DRAM controller must determine whether the 401GF is performing a burst or a non-burst access. If the DRAM controller senses /BLast high (burst access) on the rising clock edge while in state **D0**, the DRAM controller proceeds back to state **D1**, lowering /CAS and raising /Ready in the process. If the read is a non-burst access, the DRAM controller senses /BLast active and returns back to the **IDLE** state while making all the control signals inactive during the transition.

If the refresh timer expires during a DRAM access, the DRAM controller allows the 401GF to complete its access and then proceeds immediately with the refresh.



Figures 6-9 show the necessary DRAM accesses performed by the DRAM controller.





3.3 Main State Machine: Refresh Access

The MSM monitors the RCSM for a transition to its **ref_acc** state; this transition signals the MSM to perform a DRAM refresh access.

During initial power-up of the DRAM components, the MSM proceeds immediately from the **IDLE** state to **REF0** on the RCSM's first 16 transitions to **ref_acc**. On the next cycle, the MSM transitions back to **IDLE**, causing the RCSM to transition back to its **ref_delay** state. After 16 "loops" an approximate 245µs initial delay is introduced.

For all RCSM transitions to the **ref_acc** state after the 245µs delay, the MSM proceeds immediately from the **IDLE** state to the **REF3** state and activates all /CAS lines in the process. If the DRAM controller is performing a 401GF access when the RCSM jumps to its **ref_acc** state, the RCSM remains in the **ref_acc** state until the controller completes the 401GF access and acknowledges the refresh request.

The MSM transition from the **REF3** state to the **REF2** state activates the /RAS signal and the transition from the **REF2** state to the **REF1** state raises the /CAS signals back to their inactive states. An unconditional transition occurs next from **REF1** to **REF0**. The final transition from **REF0** back to the **IDLE** state raises /RAS back to its inactive state, completing the refresh access.

If during the refresh process a DRAM access from the 401GF is pending, the DRAM controller proceeds immediately from **IDLE** to state **D4** and performs a DRAM access.

Notes:

- The DSL equations found in this application note have been fully simulated and successfully fit into the MACH210-12 PLD. However, system designers should be aware that a system board has not yet been designed and tested based on this application note.
- This application note was written for the PPC401GF-MA50C2. The Address Latch Enable signal (/ALE) on the PPC401GF-MA50C2 is active low. Subsequent versions of the 401GF will incorporate an active high Address Latch Enable signal (ALE). System designers should be aware of this polarity change and design their systems accordingly.

4. PLD Equations

```
#TITLE      'DRAM Controller for the 401GF';
#ENGINEER   'Tom Aebli';
#COMPANY    'IBM PowerPC Embedded Controllers';
#REVISION   '1.0';

" INPUTS
"
INPUT
    reset,
    clk1x,
    _ale,
    ad[4],
    abus3_2[2],
    _be0,
    _be1,
    _be2,
    _be3,
    w_r,
    _blast;

" OUTPUTS
"
OUTPUT mux_cntl    CLOCKED_BY clk1x  RESET_BY  reset  DEFAULT_TO LAST_VALUE;
OUTPUT _ras0       CLOCKED_BY clk1x  PRESET_BY reset  DEFAULT_TO LAST_VALUE;
OUTPUT _cas0       CLOCKED_BY clk1x  PRESET_BY reset  DEFAULT_TO LAST_VALUE;
OUTPUT _cas1       CLOCKED_BY clk1x  PRESET_BY reset  DEFAULT_TO LAST_VALUE;
OUTPUT _cas2       CLOCKED_BY clk1x  PRESET_BY reset  DEFAULT_TO LAST_VALUE;
OUTPUT _cas3       CLOCKED_BY clk1x  PRESET_BY reset  DEFAULT_TO LAST_VALUE;
OUTPUT _ready      CLOCKED_BY clk1x  PRESET_BY reset  DEFAULT_TO LAST_VALUE;
OUTPUT ba3_2[2]    CLOCKED_BY clk1x  RESET_BY  reset  DEFAULT_TO LAST_VALUE;
OUTPUT _we         CLOCKED_BY clk1x  RESET_BY  reset  DEFAULT_TO LAST_VALUE;

" NODES
"
NODE main_states[4]    CLOCKED_BY  clk1x;
NODE idle[4]  DEFAULT_TO  0000b;
NODE d4[4]    DEFAULT_TO  0001b;
NODE d3[4]    DEFAULT_TO  0010b;
NODE d2[4]    DEFAULT_TO  0011b;
NODE d1[4]    DEFAULT_TO  0100b;
NODE d0[4]    DEFAULT_TO  0101b;
NODE ref3[4]  DEFAULT_TO  0110b;
NODE ref2[4]  DEFAULT_TO  0111b;
NODE ref1[4]  DEFAULT_TO  1000b;
NODE ref0[4]  DEFAULT_TO  1001b;

NODE ref_states[1]    CLOCKED_BY  clk1x;
NODE ref_delay[1]     DEFAULT_TO  0b;
NODE ref_acc[1]       DEFAULT_TO  1b;

NODE count_a[10]    CLOCKED_BY clk1x  RESET_BY reset  DEFAULT_TO LAST_VALUE;
NODE count_b[5]     CLOCKED_BY clk1x  RESET_BY reset  DEFAULT_TO LAST_VALUE;
```

```

NODE _ads[1]          DEFAULT_TO 1;
NODE ale_hold[1]      CLOCKED_BY clk1x  RESET_BY reset    DEFAULT_TO 0;
NODE dram_pending     CLOCKED_BY clk1x  RESET_BY reset;
NODE dram_pend_en     CLOCKED_BY clk1x  PRESET_BY reset    DEFAULT_TO LAST_VALUE;
NODE dram_en          CLOCKED_BY clk1x  RESET_BY reset    DEFAULT_TO LAST_VALUE;

" ***** "
" MAIN STATE MACHINE "
" "
" * MSB=0, LSB=31. Attention needs to be paid to which address lines on the "
" 401GF get connected to these address pins. "
" * Assumes DRAM starts at memory location 0x0XXXXXXX. "
" "
" This state machine will hold in the IDLE state until recognizing either a "
" DRAM access request from the 401GF or a refresh request initiated from the "
" from the Refresh Control State Machine. A refresh has priority over a "
" 401GF request. Both requests will indicate a pending status if either "
" occurs while the other is active. "
" "
" ***** "

STATE_MACHINE dram_sm
    STATE_BITS main_states
    CLOCKED_BY clk1x
    RESET_BY reset
    DEFAULT_TO LAST_VALUE;

STATE idle:

    IF ((ref_states = ref_acc) * (/count_b[4])) THEN
        GOTO ref0;

    ELSIF ((ref_states = ref_acc) * (count_b[4])) THEN
        _cas0 = 0;
        _cas1 = 0;
        _cas2 = 0;
        _cas3 = 0;
        GOTO ref3;

    ELSIF
        ((ref_states = ref_delay) * dram_en * ((/_ads*/ad[0]*/ad[1]*/ad[2]*/ad[3]) +
        (dram_pending))) THEN
        ba3_2 = abus3_2;
        GOTO d4;

    ELSE
        GOTO idle;

    END IF;

STATE d4:
    dram_pend_en = 0;
    _ras0 = 0;
    GOTO d3;

```

```

STATE d3:
    dram_pend_en = 1;
    mux_cntl = 1;
    IF w_r = 1 THEN
        _we = 0;
    ELSE
        _we = 1;
    END IF;
    GOTO d2;

STATE d2:
    _cas0 = _be0;
    _cas1 = _be1;
    _cas2 = _be2;
    _cas3 = _be3;
    _ready = /w_r;
    IF w_r = 0 THEN
        GOTO d1;
    ELSE
        GOTO d0;
    END IF;

STATE d1:
    ba3_2 = ba3_2 .+. 1;
    _cas0 = 1;
    _cas1 = 1;
    _cas2 = 1;
    _cas3 = 1;
    _ready = 0;
    GOTO d0;

STATE d0:
    IF (/w_r * _blast) THEN
        _cas0 = _be0;
        _cas1 = _be1;
        _cas2 = _be2;
        _cas3 = _be3;
        _ready = 1;
        GOTO d1;
    ELSIF (w_r * _blast) THEN
        _cas0 = 1;
        _cas1 = 1;
        _cas2 = 1;
        _cas3 = 1;
        _ready = 1;
        ba3_2 = ba3_2 .+. 1;
        GOTO d2;
    ELSIF (/w_r * /_blast) THEN
        _ready = 1;
        _ras0 = 1;
        mux_cntl = 0;
        GOTO idle;
    ELSIF (w_r * /_blast) THEN
        _cas0 = 1;
        _cas1 = 1;
        _cas2 = 1;
        _cas3 = 1;
        _ready = 1;
        _ras0 = 1;
        mux_cntl = 0;
        GOTO idle;
    END IF;

```



```

STATE ref_delay:

  IF (count_a[9] * /dram_en) THEN
    count_b = count_b .+. 1;
    GOTO ref_acc;
  ELSIF
    (count_a[9] * dram_en) THEN
    GOTO ref_acc;
  ELSE
    count_a = count_a .+. 1;
    GOTO ref_delay;
  END IF;

STATE ref_acc:

  IF ((main_states = ref0) * (count_b[4] /* count_b[3])) THEN
    count_a = 0;
    GOTO ref_delay;
  ELSIF
    ((main_states = ref0) * count_b[4] * count_b[3]) THEN
    count_a = 0;
    dram_en = 1;
    GOTO ref_delay;
  ELSE
    GOTO ref_acc;
  END IF;

END refresh_sm;

```

© International Business Machines Corporation, 1996

All Rights Reserved

* Indicates a trademark or registered trademark of the International Business Machines Corporation.

** All other products and company names are trademarks or registered trademarks of their respective holders.

IBM and IBM logo are registered trademarks of the International Business Machines Corporation.

IBM will continue to enhance products and services as new technologies emerge. Therefore, IBM reserves the right to make changes to its products, other product information, and this publication without prior notice. Please contact your local IBM Microelectronics representative on specific standard configurations and options.

IBM assumes no responsibility or liability for any use of the information contained herein. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of IBM or third parties. NO WARRANTIES OF ANY KIND, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE ARE OFFERED IN THIS DOCUMENT.

