EZ-030 Demonstration Board Theory of Operation

Application Note

by David Stoenner

Advanced Micro Devices

Most microprocessor evaluation and demonstration circuits are designed to illustrate the highest performance and flexibility the microprocessor can deliver, but without ease-of-design considerations. Advanced Micro Devices has developed second generation RISC microprocessors with attention to the design ease as well as the execution performance.

INTRODUCTION

The Am29030™ microprocessor provides higher performance, lower system cost, and easier design than its predecessors in the 29K™ Family. The standalone EZ-030 demonstration board displays the ease of design and low-cost system features of the Am29030 microprocessor, producing an elegant high-performance low-complexity design. This application note explores the theory of operations and design considerations of the EZ-030 demonstration board.

The EZ-030 demonstration board circuit is designed to demonstrate the simplicity of a Page Mode DRAM memory system to support the Am29030 microprocessor. Although the circuit is simple and flexible, it does not sacrifice execution performance. The EZ-030 demonstration board performance is not restricted by its DRAM controller.

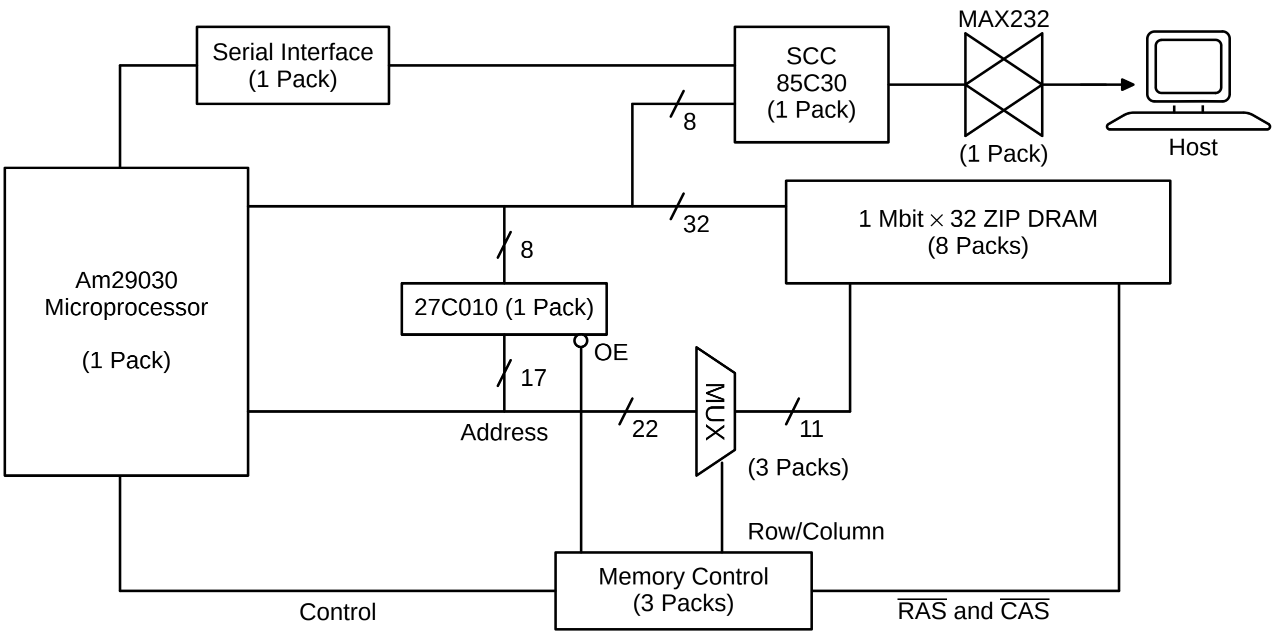
The EZ-030 demonstration board has four functional sections (see Figure 1):

- the Am29030 microprocessor and supporting parts
- the memory and memory control components
- the UART and RS232 support
- the power-on reset chip and push button

All of the above circuitry fits into 14 square inches of board space (16 in^2 including connectors and standoffs).

Am29030 MICROPROCESSOR

The Am29030 microprocessor is a 32-bit RISC microprocessor executing the 29K Family instruction set. The predecessor of the Am29030 microprocessor, the Am29000® microprocessor has a modified Harvard three-bus architecture with a single address bus multiplexed for its instruction and data buses. The Am29030 microprocessor has a simplified two-bus architecture (Address bus and Instruction/Data bus) and a large on-chip I-cache (8 Kbyte).



17580A–001

Figure 1. EZ-030 Demonstration Board Block Diagram

Publication# **17580** Rev. **A** Amendment **/0**
Issue Date: **January 1994**

The first generation of AMD signal-compatible embedded RISC microprocessors are the Am29000, Am29005™, and Am29050™ microprocessors. Each of these microprocessors has an external modified Harvard architecture to expose the internal instruction and data buses. The system designer can create two independent memory systems to provide necessary bandwidth for a design. The Am29000 microprocessor provides the highest price/performance value of any RISC microprocessor.

The second generation of the 29K Family extends performance while reducing design complexity and design-in effort. The Am29030 and Am29035™ microprocessors extend the range of price/performance beyond the Am29000 processor. The Am29030 microprocessor bus has reduced the signal complement, relaxed the timing requirements, and added special memory support to reduce system cost, making it a designer-friendly part.

Clock

The Am29030 microprocessor clock timing is forgiving and flexible. For example, the CPU can tolerate an input clock duty cycle of 40 percent to 60 percent. The Am29030 microprocessor also has Scalable Clocking™ technology that runs the internal processor core at twice the bus frequency. The EZ-030 demonstration board uses this Scalable Clocking feature. Using the configuration jumpers on the EZ-030 demonstration board, the designer can allow for a double frequency or 32-MHz input clock (INCLK) while maintaining a 16-MHz external memory clock (MEMCLK). Because only the PGA packages of the Am29030 can dissipate the heat reliably at 25 and 33 MHz, their timing is used for analysis in this Application Note. An Addendum at the end of this Application Note shows how the Cerquad 16-MHz Am29035 can be used in this memory environment. Running the processor at 2X frequency provides nearly 1.5 times the performance of a 1X INCLK, even though the external memory bus runs at 1X. This feature allows the EZ-030 demonstration board to get the most performance for its DRAM bandwidth.

A side benefit of using the Scalable Clocking feature with the internally derived MEMCLK is relaxed interface timing restrictions (see Table 1). Processor outputs are valid closer to the rising edge of MEMCLK and input data has less required setup time.

Interface

The EZ-030 demonstration board exploits features of the Am29030 microprocessor to derive the maximum performance for the minimum cost and complexity. The first feature the EZ-030 demonstration board uses is the Am29030 microprocessor's narrow read feature for EPROM.

By asserting the read-narrow pin ($\overline{\text{RDN}}$) during an EPROM access of the Am29030 microprocessor, the Am29030 microprocessor reads instructions one byte at a time and generates an address for each byte. The instruction data is read from the least significant byte (bits 31 to 24) of the bus, because the EZ-030 demonstration board is a big-endian design.

On the EZ-030 demonstration board, an 8-bit EPROM is attached with the read narrow interface. This EPROM contains MiniMON29K™ code, used for bootstrap code to download programs into DRAM. The EPROM also contains debugger support and host interface (HIF) support. HIF allows the host operating-system standard I/O services, such as open, close, read and write file, and others, to be passed through the EZ-030 demonstration board to the host.

Table 1. PGA Interface Timing Restrictions[1]

No.	Parameter Description	Test Conditions	Preliminary 25/33 MHz Min	Preliminary 25/33 MHz Max	Unit
7	Synchronous Output Valid Delay	MEMCLK Output MEMCLK Input	1 2	13 17	ns
7a	Synchronous Output Valid Delay for ID31–ID0	MEMCLK Output MEMCLK Input	1 2	13 17	ns
9	Synchronous Input Setup Time	MEMCLK Output MEMCLK Input MEMCLK=INCLK	17 17 12		ns
9a	Synchronous Input Setup Time for ID31–ID0	MEMCLK Output MEMCLK Input MEMCLK=INCLK	9 9 6		ns

Notes:

[1] *Subject to change without notice. Refer to the "Am29030 and Am29035 Microprocessors Data Sheet" for a complete listing of AC/DC parameters (order# 18150).*

The Am29030 microprocessor is very well suited for page-mode or static column DRAM. The microprocessor takes advantage of page-mode memory systems to provide single-cycle accesses to same page requests. The Am29030 microprocessor has a special signal to notify the memory system that accesses are sequential. These sequential accesses occur for cache line fills, and multiple loads and stores of registers. The EZ-030 card utilizes only the sequential $\overline{\text{BURST}}$ signal for its page mode implementation.

Sequential accesses are marked by the $\overline{\text{BURST}}$ signal of the Am29030 microprocessor. The $\overline{\text{BURST}}$ signal never spans DRAM page boundaries; in fact, the burst access does not cross 1-Kbyte memory boundaries. The 1K boundary is known as a processor page boundary, because the processor checks the Memory Management Unit (MMU) at this boundary for page faults. If the processor MMU is not activated, this boundary is still respected. Thus, page-mode and static column DRAMs snap in with no need for special address comparators.

The Am29030 microprocessor furnishes a new address during each burst access cycle, which can be “strobed” into the page-mode DRAM by the $\overline{\text{CAS}}$ strobe pulsing at an appropriate time. To generate this pulsing $\overline{\text{CAS}}$ strobe, the EZ-030 demonstration board uses a delay line. The delay line produces a delayed version of MEMCLK. The real MEMCLK and the delayed version are “ANDed” together to produce the CAS pulse train used during the $\overline{\text{BURST}}$ signal. (The Am29030 microprocessor has on-chip page comparators and interleave support features that the EZ-030 demonstration board does not use.)

JTAG Connector

The EZ-030 demonstration board has a standard 10-pin JTAG connector to access the Am29030 microprocessor part on the board. The JTAG IEEE scan standard provides low-cost emulation and simpler in-circuit testing. Using this connector, it is possible to access every pin of the Am29030 microprocessor through the five JTAG pins. Because the JTAG reset pin must be tied to the reset of the processor when the JTAG port is not in use, a jumper must be disconnected when a JTAG device is attached to the EZ-030 demonstration board. See the *EZ-030 Demonstration Board Reference Manual* for details.

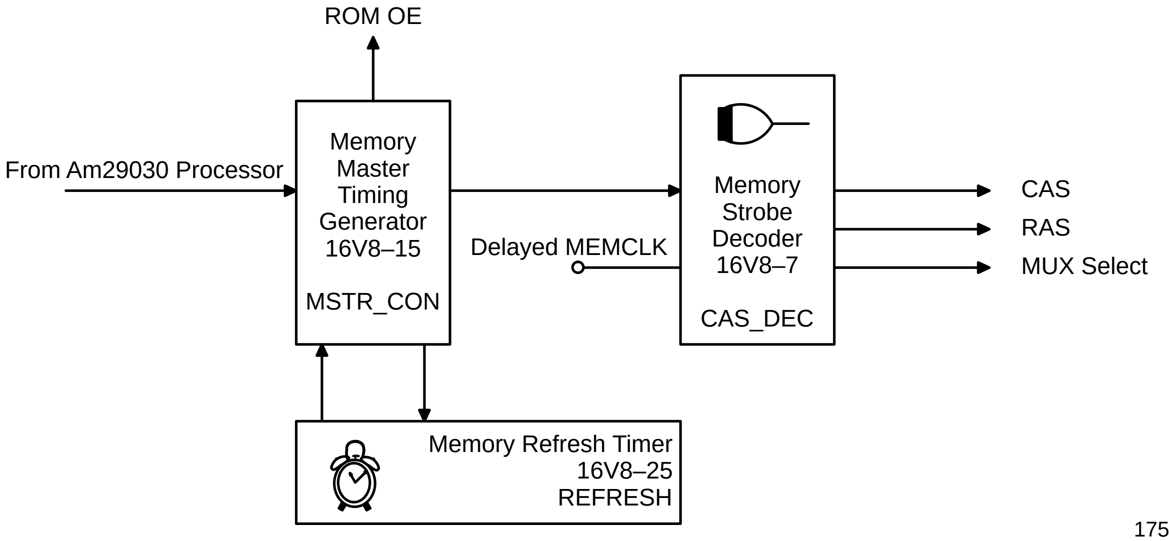
MEMORY DESCRIPTION

The memory system of the EZ-030 demonstration board is composed of three PAL® devices, three 74LS157 MUXs, one EPROM (27C010), and eight ZIP DRAMs (256K x 4–70 or 1 Mbyte x 4–70). The three PAL ICs that form the memory controller of the EZ-030 demonstration board are the master control PAL device (MSTR_CON), which is a PALACE16V8–15, the RAS/CAS decode PAL device (CAS_DEC), which is a PAL16V8–7, and the refresh timer PAL device (REFRESH), which is a PALACE16V8–25 (see Figure 2).

STATE MACHINE DESCRIPTION

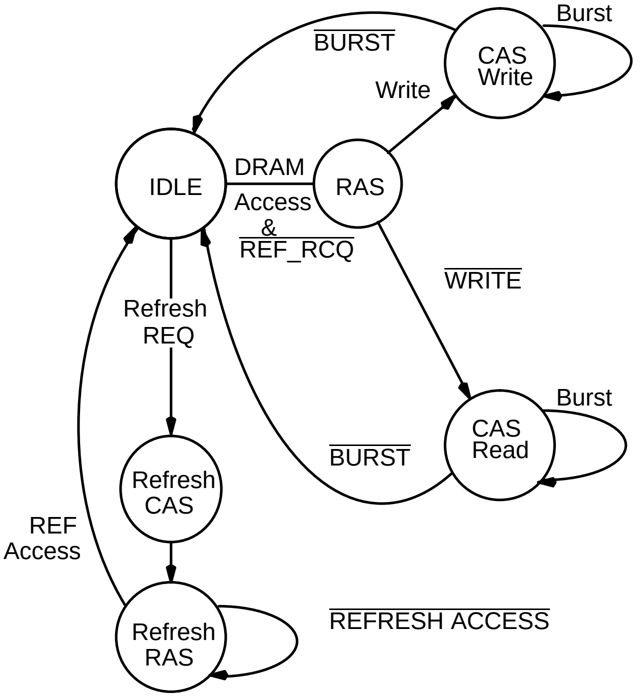
Figure 3 is a state machine diagram of the EZ-030 memory system. The ready signal, $\overline{\text{RDY}}$, is the termination of any access. Since the microprocessor samples the $\overline{\text{RDY}}$ signal to indicate the completion of an access, $\overline{\text{RDY}}$ can be supplied by the serial communications control PAL ICs or the memory control PAL ICs. Below is a description of the $\overline{\text{RDY}}$ derivation. The state machine represents the primary states and state transitions. These are IDLE, Row Address ($\overline{\text{RAS}}$), Burst READ, Burst WRITE, Refresh CAS, and REFRESH RAS.

The state machine trajectories create the following: (1) three-cycle random accesses to DRAM, (2) one-cycle sequential (BURST) accesses to DRAM, and (3) three-cycle refresh. The following sections describe the PAL equation source files.



17580A–002

Figure 2. Memory System Block Diagram


Burst
$\overline{\text{BURST}}$
CAS Write
Write
IDLE
DRAM Access & $\overline{\text{REF\_RCQ}}$
RAS
$\overline{\text{WRITE}}$
Refresh REQ
Burst
CAS Read
$\overline{\text{BURST}}$
Refresh CAS
REF Access
Refresh RAS
$\overline{\text{REFRESH ACCESS}}$

17580A–003

Figure 3. DRAM State Machine Diagram

READY SIGNAL GENERATION

MSTR_CON is the timing generator for the memory system. When the Am29030 microprocessor asserts the access request signal $\overline{\text{REQ}}$, the MSTR_CON PAL device produces a ready signal ($\overline{\text{RDY}}$) [1.2a]. The $\overline{\text{RDY}}$ signal is a combinatorial signal. That is, it is a combination of the ready terms for both memory mapped devices, the serial port, and the memory system. The memory ready signal, MEM_RDY, is a registered signal as shown by the ":=" equate symbol [1.1a].

$$
\begin{aligned}
&\text{MEM\_RDY} := \text{REQ}\overline{\text{*A31*A30*MEM\_RDY}} && \text{[1.1a]}\\
&\quad + \text{MEM\_ACCESS}\overline{\text{*MEM\_RDY}} && \text{[1.1b]}\\
&\quad + \text{MEM\_ACCESS*MEM\_RDY*BURST} && \text{[1.1c]}\\
&\text{RDY} = \text{MEM\_RDY} + \text{SERIAL\_RDY} && \text{[1.2a]}
\end{aligned}
$$

Equation 1. MSTR_CON PAL Equations

The MSTR_CON PAL device has MEMCLK as its clocked input. If there is a microprocessor access request while the most significant address lines, A31 and A30, are zero (EPROM address space), the MEM_RDY [1.1a] signal activates in the next cycle. The Am29030 microprocessor does not sample the $\overline{\text{RDY}}$ signal until the next cycle. This results in a two-cycle access, one cycle for the MSTR_CON PAL device to sample the request and one cycle for the Am29030 microprocessor to sample the $\overline{\text{RDY}}$ signal. For subsequent EPROM accesses, the ready signal goes active for one cycle and then inactive for the next cycle, continuing two-cycle timing.

During a DRAM access, MEM_ACCESS must be active before MEM_RDY activates [1.1b]. MEM_RDY is the memory system component of the RDY signal. The next section shows that MEM_ACCESS is a registered signal that indicates a DRAM access was requested. Since MEM_ACCESS is a registered signal, the RDY for DRAM is delayed one more cycle than the EPROM, or three cycles total. These three cycles are: one cycle for MEM_ACCESS to go active, one cycle for MEM_RDY to go active (which triggers the processor $\overline{\text{RDY}}$ signal in the same cycle), and one cycle for the RDY signal to be sampled by the microprocessor.

During the burst access of the Am29030 microprocessor, the $\overline{\text{RDY}}$ signal is held active as long as $\overline{\text{BURST}}$ is active [1.1c]. This provides a single-cycle access acknowledge because $\overline{\text{RDY}}$ is sampled every cycle. The EPROM does not support $\overline{\text{BURST}}$.

DRAM TIMING

The IDLE signal represents the time when the DRAM system is between accesses. The 70-ns DRAM has a RAS precharge time of 50 ns [2.1a]. Then, the MSTR_CON PAL device requires a visit of the IDLE state before it can go to the RAS state and then onto the CAS state. This visit takes at least one cycle which is 63 ns at 16 MHz. While $\overline{\text{IDLE}}$ is asserted, $\overline{\text{RAS}}$ is deasserted. Therefore, the $\overline{\text{RAS}}$ strobe remains inactive for 63 ns, which is long enough to meet the precharge duration requirement.

MEM_ACCESS is a signal that indicates a DRAM access is in progress. IDLE is a signal that indicates DRAM is inactive. IDLE asserts when no DRAM activity is detected, specifically, when there is no DRAM access and no refresh access ($\overline{\text{MEM\_ACCESS}}$*$\overline{\text{REF\_ACCESS}}$) [2.1a]. IDLE is reinforced if: no access is requested to

$$
\begin{aligned}
&\text{IDLE} := \overline{\text{IDLE}}\text{*}\overline{\text{MEM\_ACCESS*REF\_ACCESS}} && \text{[2.1a]}\\
&\quad + \text{RESET} && \text{[2.1b]}\\
&\quad + \overline{\text{IDLE}}\text{*MEM\_ACCESS*ST1*MEM\_RDY*}\overline{\text{BURST}} && \text{[2.1c]}\\
&\quad + \overline{\text{IDLE}}\text{*REF\_ACCESS*}\overline{\text{REF\_REQ}} && \text{[2.1d]}\\
&\quad + \text{IDLE*(}\overline{\text{REQ*A31*/A30}}\text{)*}\overline{\text{REF\_REQ}}\text{*}\overline{\text{EMUL}} && \text{[2.1e]}\\
&\text{REF\_ACCESS} := \text{IDLE*REF\_REQ*}\overline{\text{REF\_ACCESS}} && \text{[2.2a]}\\
&\quad + \text{REF\_ACCESS*REF\_REQ} && \text{[2.2b]}\\
&\text{MEM\_ACCESS} := \text{IDLE*REQ*A31*}\overline{\text{A30*REF\_REQ}}\text{*}\overline{\text{MEM\_ACCESS}}\text{*EMUL} && \text{[2.3a]}\\
&\quad + \text{MEM\_ACCESS*}\overline{\text{ST1}} && \text{[2.3b]}\\
&\quad + \text{MEM\_ACCESS*BURST} && \text{[2.3c]}\\
&\text{ST1} := \text{MEM\_ACCESS*}\overline{\text{MEM\_RDY}} && \text{[2.4a]}\\
&\quad + \text{ST1*BURST} && \text{[2.4b]}
\end{aligned}
$$

Equation 2. MSTR_CON PAL Equations

$$
\begin{aligned}
\text{RAS0} &= \text{MEM\_ACCESS} && [3.1a]\\
&+ \text{REF\_ACCESS}*\overline{\text{MEMCLK}} && [3.1b]\\
&+ \text{RAS0}*\text{REF\_ACCESS} && [3.1c]\\
\\
\text{MUX} &= \text{MEM\_ACCESS}*\overline{\text{MEMCLK}} && [3.2a]\\
&+ \text{MUX}*\text{MEM\_ACCESS} && [3.2b]\\
\\
\text{CAS0} &= \text{REF\_ACCESS} && [3.3a]\\
&+ \text{MUX}*\text{MEM\_ACCESS}*\text{ST1}*\overline{\text{WRITE}}*\text{MEMCLK}*\\
&\quad \text{DELAY\_IN} && [3.3b]\\
&+ \text{MUX}*\text{MEM\_ACCESS}*\text{ST1}*\overline{\text{WRITE}}*\text{CAS0}*\\
&\quad \text{DELAY\_IN} && [3.3c]\\
&+ \text{MUX}*\text{MEM\_ACCESS}*\text{ST1}*\overline{\text{WRITE}}*\text{CAS0}*\\
&\quad \overline{\text{MEMCLK}} && [3.3d]\\
&+ \text{MUX}*\text{MEM\_ACCESS}*\text{ST1}*\text{WRITE}*\text{WE0}*\\
&\quad \overline{\text{MEMCLK}} && [3.3e]
\end{aligned}
$$

Equation 3. CAS_DEC PAL Equations

DRAM (2 MSBs A31, A30 = 10_2) [2.1e], no refresh request is pending ($\overline{\text{REF\_REQ}}$), and no emulator accesses are attempted ($\overline{\text{EMUL}}$). IDLE is asserted if a non-burst (simple) access is complete: (MEM_ACCESS*ST1*MEM_RDY*$\overline{\text{BURST}}$) [2.1c]. IDLE is also asserted if a refresh has completed: (REF_ACCESS*$\overline{\text{REF\_REQ}}$) [2.1d].

REF_ACCESS is a signal that acknowledges the REF_REQ when the DRAM is idle [2.2a]. The REF_ACCESS releases after the REF_REQ is released, acknowledging the completion of a refresh [2.2b]. The refresh operation can occur while EPROM or Serial Port accesses are also occurring.

MEM_ACCESS is initiated when the Am29030 microprocessor accesses DRAM space, where A31=1 and A30=0 [2.3a]. This is qualified by no refresh request and no emulator access. MEM_ACCESS persists for one more cycle until ST1 is asserted. MEM_ACCESS is reinforced by burst which might take from 1 cycle to 255 cycles. Then MEM_ACCESS waits for an IDLE cycle. Since MEM_ACCESS creates RAS, the IDLE cycle is the RAS precharge delay. MEM_ACCESS creates RAS fairly directly through the CAS_DEC PAL device.

The ST1 signal is used to flag the last cycle of a memory access [2.3b]. The CAS_DEC PAL device decodes the ST1 signal to form the CAS memory strobes for the proper memory devices [2.4a]. During a write, only the bytes with WE strobes result in CAS strobes. This is how partial words (bytes and half-words) are stored. The ST1 signal persists during burst accesses [2.4b].

The CAS_DEC PAL Device

The CAS_DEC PAL device is a PAL device 16V8–7. The row and column address strobes are decoded by the CAS_DEC PAL equations. Additionally, the strobe signal, MUX, which chooses column address instead of row address, is decoded out of this PAL device. See Equation 3 for the equations for RAS, CAS, and the row and column address strobes.

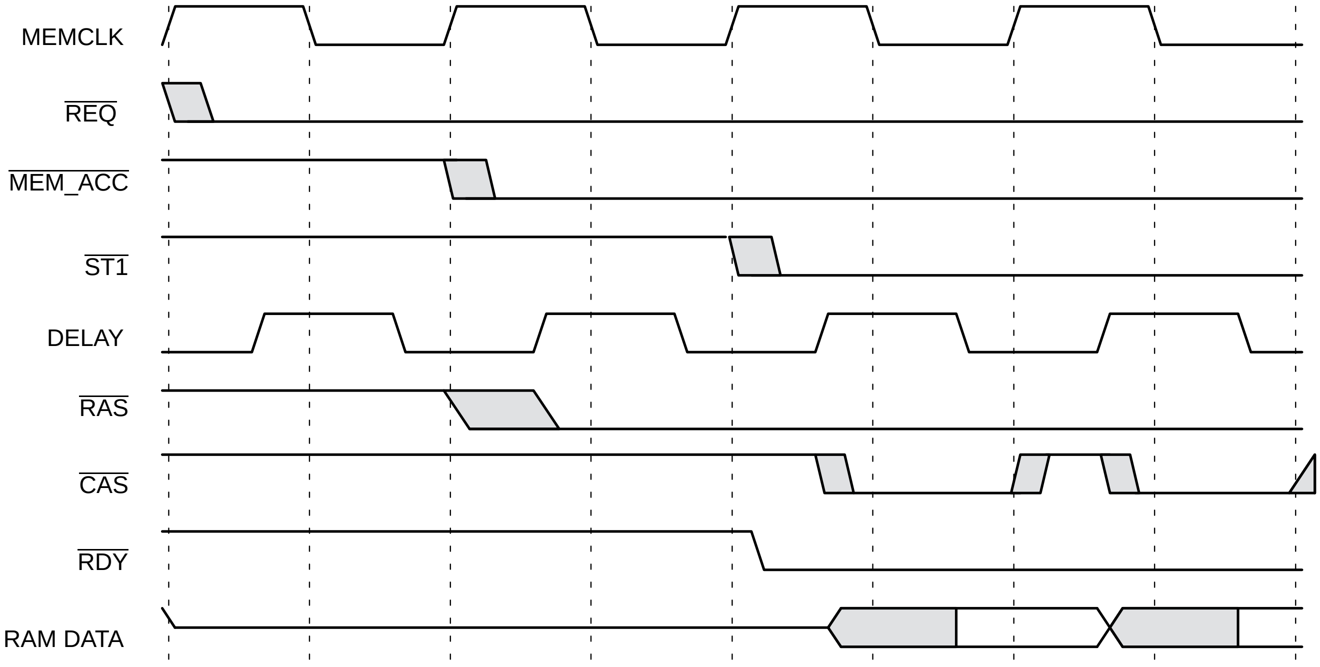
The RAS0 strobe is used by all eight ZIP DRAM chips to clock in the ROW address. RAS0 is derived from MEM_ACCESS of the MSTR_CON PAL device. MEM_ACCESS indicates the beginning of a memory cycle and RAS initiates the cycle [3.1a]. The additional two terms of RAS0 create the pulse for CAS before RAS refresh. The $\overline{\text{RAS}}$ signal is delayed half a MEMCLK from the onset of REF_ACCESS [3.1b].

The MUX [3.2a and 3.2b] signal connects to the A/B control input of the three 74LS157 two-input MUXs. This signal is used to choose between the row addresses and the column addresses. The MUX signal chooses the column address half a MEMCLK cycle after the $\overline{\text{RAS}}$ strobe activates. This first-half clock cycle gives the row addresses 30 ns of hold time. The required time is 15 ns for 70-ns DRAMs. The second-half MEMCLK cycle allows the column address to propagate through the 74LS157 MUX. This delay signal provides the MUX switching time. The switching delay of the 74LS157 MUX is 27 ns, and the half-cycle plus the delay signal is 50 ns. Both the MSTR_CON and the CAS_DEC PAL devices allow additional setup time tolerance. The setup time for the column address on a 70-ns DRAM is 0.

Only one of the CAS equations is shown here [3.3a–3.3e] because the rest of the CAS signals CAS1–CAS3 are symmetrical. The first term of the CAS equation is the CAS before RAS refresh. CAS becomes active as soon as a refresh cycle begins. The REF_ACCESS signals the beginning of the refresh cycles. The second, third and fourth terms are for normal memory accesses. The $\overline{\text{CAS}}$ strobe for the first read or simple read cycle to DRAM occurs in the last of two cycles. The delay signal (DELAY_IN), is MEMCLK passed through a 20-ns delay line. The $\overline{\text{CAS}}$ strobe continues until MEMCLK becomes active in the last equation.

The delay signal is necessary for the burst read access that might follow the simple access. A burst read access starts at the end of a simple access. The second through fourth terms of the CAS equation shape the CAS pulse to provide the appropriate setup time for the column address [3.3b–3.3d].

The fourth term [3.3d] allows the assertion of MEMCLK to deassert any previous CAS. The first term [3.3a] takes effect, allowing a delay of DELAY_IN (20 ns) before CAS becomes active. These two terms have now provided a 20-ns pulse that meets the 10-ns CAS precharge specification of a 70-ns DRAM. The 20-ns pulse of CAS inactive allows the 14-ns address propagation delay through the 74LS157 MUXs. The second term carries the $\overline{\text{CAS}}$ signal into the third term. DELAY_IN overlaps MEMCLK by 20 ns, so that CAS is reinforced by the second term until DELAY_IN goes inactive (see Figure 4 and Figure 5). The page-mode access time from CAS active to data accessed is 20 ns, which allows the microprocessor 15 ns of setup time (9 ns is the requirement).



17580A–004

Figure 4. DRAM READ Timing Diagram

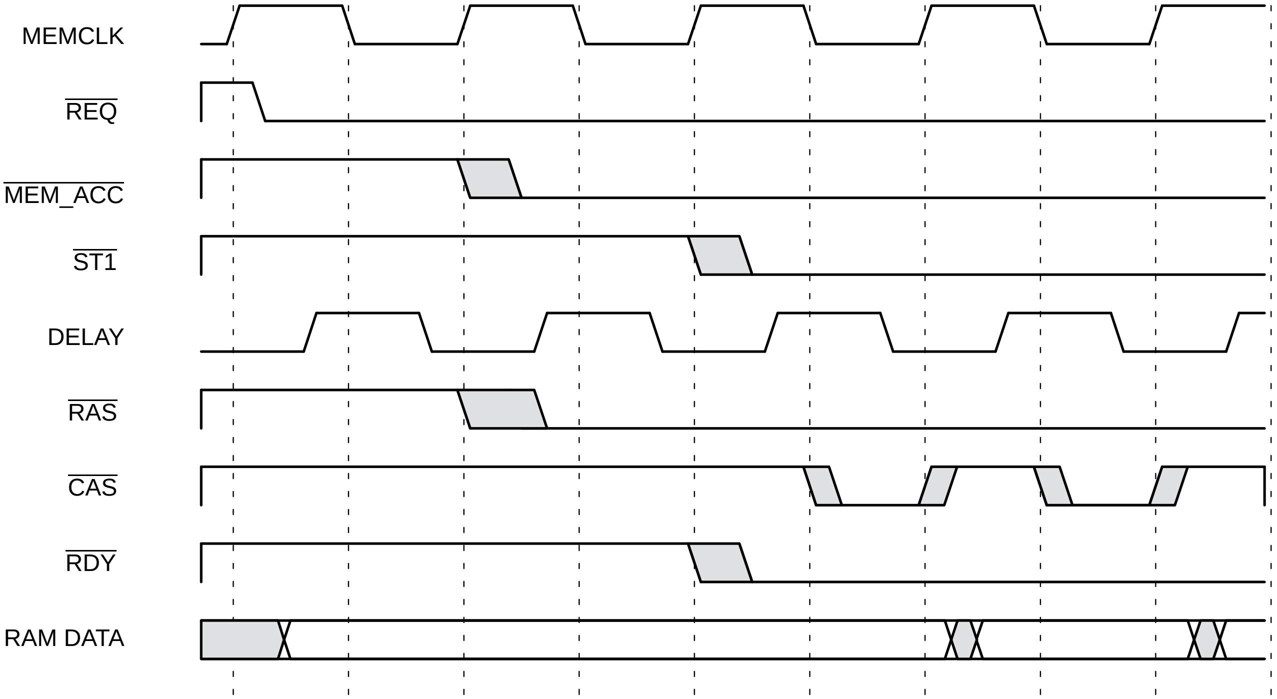


Figure 5. DRAM Write Timing Diagram

17580A–005

The single-cycle write CAS is easier than the single-cycle read CAS, because the data is available near the beginning of the cycle. All that is necessary is a $\overline{\text{CAS}}$ strobe at the back half of the cycle after the column address is set up. The equation term:

MUX*MEM_ACCESS*ST1* WRITE *WE0*$\overline{\text{MEMCLK}}$ [3.3e]

activates CAS signal to become active as MEMCLK (i.e., it is active during the back half of MEMCLK). The remaining signals in this term are conditions for the CAS pulse(s) to occur. Note the WE0 signal would be WE1, WE2, or WE3 for CAS1, CAS2, or CAS3. This WE signal is a byte-write strobe signal from the Am29030 microprocessor. Only the CAS lines of the microprocessor-requested bytes are written.

Refresh Timer

The refresh timer logic is a simple counter with two exposed signals, REF_REQ and REF_ACCESS. REF_REQ is the request generated every 129 clocks. REF_ACCESS is the acknowledge signal that a refresh access is in progress. Refresh access only clears the REF_REQ signal. Therefore, refreshes can be pending for up to 129 cycles before they are lost (see Equation 4).

Q0 := $\overline{\text{Q0}}$
Q1:=Q1:+: Q0 [4.1]
Q2:=Q2:+: (Q1*Q0) [4.2]
Q3:=Q3:+: (Q2*Q1*Q0) [4.3]
...
Q6:=Q6:+: (Q5*Q4 *Q3*Q2*Q1*Q0) [4.4]
REF_REQ:= Q6*Q5*Q4*Q3*Q2*Q1*Q0 [4.5a]
+REF_REQ*$\overline{\text{REF\_ACCESS}}$ [4.5b]

Equation 4. REFRESH PAL Equations

EPROM

The ROM_CS signal is routed to both the $\overline{\text{RDN}}$ and to the EPROM (27C010) chip-select. Because of the read-narrow feature of the Am29030 microprocessor, the EPROM accesses are easy to facilitate. There is one trick though—the read-narrow signal ($\overline{\text{RDN}}$) of the Am29030 microprocessor must be driven during a processor reset to indicate whether an 8-bit or a 16-bit EPROM is used. For an 8-bit EPROM, the $\overline{\text{RDN}}$ signal must be driven inactive (High) [5.1]. This is the default state of the EPROM chip-select (ROM_CS) signal from the MSTR_CON PAL device.

The ROM_CS equation only produces a chip-select and $\overline{\text{RDN}}$ during a request to the EPROM address space. The EPROM address space is any address with the MSBs, A31 and A30, both equal to zero [5.1]. Note that when a processor is reset, execution starts from location zero, which is in the EPROM address space.

ROM_CS = $\overline{\text{EMUL}}$*REQ*$\overline{\text{A31}}$*$\overline{\text{A30}}$ [5.1]

Equation 5. MSTR_CON PAL Equation

SERIAL PORT SUPPORT

The Serial Communications Controller (SCC) chip is attached to the most significant byte of the Am29030 microprocessor. To provide 5 V-only operation of RS232 drives, a MAX232 chip is used to create ±10 V level signals for the receive data, the transmit data, the Data Terminal Ready, and Data Set Ready signals. The connector is configured so that the EZ-030 demonstration board is Data Communications Equipment (DCE) and attaches to a host, such as a PC-AT that is Data Terminal Equipment (DTE). The EZ-030 debug monitor serial protocol requires an intelligent host.

The Serial Port section consists of the Am85C30 chip, the serial controller PAL device (SERIAL), the MAX232 chip, a baud rate oscillator, and the subminiature 9-pin D connector.

The Am85C30 Chip

The Am85C30 chip is known as the Serial Communications Controller (SCC). It is a dual-channel, multi-protocol data communications peripheral chip. The SCC can be configured to provide asynchronous SDLC and HDLC interface. This device can check and generate CRC codes.

The EZ-030 circuit with standard target Universal Debugger Interface (UDI) target monitor software utilizes a very small subset of the functionality of this SCC part. The EZ-030 board has a connector for only one of the ports on the Am85C30 chip. The SCC is connected to an independent 2.4576-MHz oscillator for baud rate precision.

The SCC Interface PAL Device

The SERIAL PAL device creates the signals to interface the Am29030 microprocessor to the Am85C30 chip. SERIAL is clocked by MEM_CLK. The 8530_CS is used for the chip enable (CE) to the SCC [6.1]. One timing specification on the Am85C30 chip is enforced by software. Because the SCC is run by an independent oscillator, a circuit inside the part prevents metastability by multiply-sampling source data. Because of this resampling delay, accesses to the chip must be spaced by six clocks of the SCC oscillator plus 300 ns.

To accommodate this extended timing spec, the EZ-030 software accessing the SCC directly must delay between accesses. When computing this delay, the cache and double internal frequency considerations must be accounted for. For the 2.43567-MHz oscillator attached to a 16-MHz internal frequency part, the number of delay cycles between accesses must be 45 cycles.

$$\begin{aligned}
&\text{8530\_CS = REQ*A31*A30} && [6.1]\\
&\text{SERIAL\_RDY := NC1*NC2*}\overline{\text{SERIAL\_RDY}} && [6.2]\\
&\text{NC1 := (8530\_RD + 8530\_WR)*}\overline{\text{NC1*SERIAL\_RDY}} && [6.3]\\
&\text{NC2 := (8530\_RD + 8530\_WR)*(NC1:+:NC2)} && \\
&\qquad\text{*}\overline{\text{SERIAL\_RDY}} && [6.4]
\end{aligned}$$

Equation 6. SCC Interface PAL Equations

$$\begin{aligned}
&\text{8530\_RD:=8530\_CS*}\overline{\text{WRITE}}\text{*}\overline{\text{8530\_RD}}\text{*}\overline{\text{SERIAL\_RDY}} && [7.1a]\\
&\qquad\text{+ 8530\_RD*}\overline{\text{SERIAL\_RDY}} && [7.1b]\\
&\qquad\text{+ RESET} && [7.1c]\\
&\text{8530\_WR := 8530\_CS*WRITE*}\overline{\text{8530\_WR}} && \\
&\qquad\text{*}\overline{\text{SERIAL\_RDY}} && [7.2a]\\
&\qquad\text{+ 8530\_WR*}\overline{\text{NC1}}\text{*}\overline{\text{SERIAL\_RDY}} && [7.2b]\\
&\qquad\text{+ 8530\_WR*}\overline{\text{NC2}}\text{*}\overline{\text{SERIAL\_RDY}}\text{ + RESET} && [7.2c]
\end{aligned}$$

Equation 7. SCC Interface PAL Equations

The SERIAL_RDY signal is a term used by the MSTR_CON PAL device [1.2a] to provide the $\overline{\text{RDY}}$ signal to the Am29030 microprocessor [6.2]. The NC1 and NC2 (no connect 1 and 2) terms form a counter to provide the proper delay for reads and writes to allow transfer [6.3] and [6.4]. The resulting access is active five cycles: one cycle for NC1, one cycle for NC2, one cycle for NC1 to be active again, one cycle for SERIAL_RDY to sample NC1 and NC2, and one cycle for the RDY to be sampled by the Am29030 microprocessor. Note that the software timing restriction allowing six SCC oscillator cycles between accesses is over and above the NC1 and NC2.

The reset signal causes a simultaneous read and write signal to the Am85C30 chip, which indicates a reset to the SCC [7.1c] and [7.2c]. This can be done by software as well. The SERIAL_RDY signal is initiated one cycle after the 8530_RD signal, which begins one propagation delay after the Am29030 microprocessor access to the serial port. The 8530_RD signal lasts until the SERIAL_RDY is done [7.1b]. The 8530_WR write pulse lasts for three clocks, bracketed by one clock cycle on either side [7.2a]. (This is for the data hold time requirement of the Am85C30 chip. Note that a slower than 8-MHz part requires an extra wait state here.)

FINISHING TOUCHES

The EZ-030 demonstration board uses a TL7705 chip to provide a clean, “debounced,” power-on reset signal and push-button reset signal. It is possible to attach a jumper between the DTR signal and the push button to allow host-controlled reset serial port. The RESET signal configures the processor for 8-bit operation and resets the SCC and the Am29030 microprocessor.

The 74LS157 MUXs have long propagating delays and switching times, and therefore do not require series termination resistors for the highly capacitive eight fan-out DRAM address bus. If high-speed MUXs or PAL devices are used in lieu of the 74LS157 MUXs, it is recommended that series 33-ohm termination resistors be used to moderate pulse switching times and dampen reflections.

ADDENDUM FOR 16-MHZ AM29035 CERQUAD

This application note presented how to make a simple memory interface for the Am29030, but the timing was for the PGA package parts that, because of the superior power distribution, can have the better timings shown in Table 1. The Cerquad parts, though, have different timings than their PGA counterparts, as shown in Table 2. This needs to be taken into account in the worst-case analysis of the design. This Addendum shows how to produce a design for the Cerquad parts that can still maintain the performance of the design shown previously.

The problem comes in worst-casing the address to the CAS control path to meet the ID31–ID0 setup. The CAS term cannot be lowered until the address is valid at the output of the multiplexers (74LS157's). The equation is:

$$\begin{array}{l} T_7 + \text{74LS157} - \text{(16V8–7) PAL}_{\text{fastest delay}} \\ 22 + 14 - 2 = 34 \text{ ns} \end{array}$$

making the delay line 34 ns instead of 20 ns.

This, then, adds to the CAS an access of 20 ns plus the worst-case delay time of the PAL device, plus the setup of the Am29035. This equation becomes:

$$34 \text{ ns} + 20 \text{ ns} + 7 \text{ ns} + 17 \text{ ns} \Rightarrow 78 \text{ ns}$$

This 78 ns significantly exceeds the 63 ns allowed so modifications must be made to the design.

There are two paths to fix the design. Either fix the address clock to Q delay, or fix the data setup side. To fix the data setup, a register with a better setup, say a 74F374 with a setup of 4 ns, can be inserted. This solves the problem, but adds an extra clock to each access. This also adds four parts to the design for the full 32-bit data bus. There is a less expensive solution. Since the only problem is on CAS and not RAS, only eight address lines need to be dealt with instead of 32 data lines. A 20V8–10-ns PAL device can be used as a clock-enabled register with a control-enabled count. This then reduces the equivalent of T_7 from 22 ns to 7 ns. Along with using a 5-ns PAL device for CAS decode, this results in the setup time required with –1 ns of margin. To fix the –1 ns of margin, the delay line should be reduced to 15 ns and the 74LS157s changed to 74F157s. This results in an access equation of:

$$\begin{array}{l} \text{Delay line + PAL delay + CAS access + setup} \le 63 \\ 15 + 5 + 20 + 19 = 59 \le 63 \text{ ns} \end{array}$$

The equation for this PAL device (ADD_CNTR.PDS) is given at the end of the PAL equation section. This PAL device should fit between the intersection of the address bus for the PROM and the multiplexers, as shown in Figure 1, and is only needed on the lower address bits A9–A2. There are two control terms on this PAL device load, which cause the register to be updated with the address and INC, which in turn cause the register to count upward. In this design, the term IDLE can be used in the master control PAL device to drive LOAD. The address bus is valid 1 clock before IDLE becomes false, so the register is updated with the correct address and held. Holding this address until the first CAS pulse enables the ST1 to be used as the needed INC term to produce counts in the burst sequence.

These modifications to the basic memory system make the significantly cheaper Cerquad part function at or near its top speed. Only one small change is needed to the EPROM. The speed of the EPROM must be decreased to 70 ns instead of the 90 ns as used in the 25- and 35-MHz designs.

Table 2. 16-MHz Am29035 Cerquad Interface Timing Restrictions[1]

No.	Parameter Description	Test Conditions	Preliminary 16 MHz Min	Preliminary 16 MHz Max	Unit
7	Synchronous Output Valid Delay	MEMCLK Output MEMCLK Input	1 2	20 24	ns
7a	Synchronous Output Valid Delay for ID31–ID0	MEMCLK Output MEMCLK Input	1 2	22 26	ns
9	Synchronous Input Setup Time	MEMCLK Output MEMCLK Input MEMCLK=INCLK	21 21 17		ns
9a	Synchronous Input Setup Time for ID31–ID0	MEMCLK Output MEMCLK Input MEMCLK=INCLK	17 17 13		ns

Notes:

[1] *Subject to change without notice. Refer to the "Am29030 and Am29035 Microprocessors Data Sheet" for a complete listing of AC/DC parameters (order# 18150).*

APPENDIX A. PAL EQUATIONS

This appendix contains PAL equations for the EZ-030 demonstration board.

```
;PALASM® Design Description
;--------------- Declaration Segment ---------------
TITLE      RAS AND CAS GENERATOR FOR EZ-030
PATTERN    CAS_DEC.PDS
REVISION   A
COMPANY    AMD
DATE       03/03/92

CHIP  U3  PAL16L8 (or PALCE16V8)

;--------------- PIN Declarations ---------------
PIN  1            MEMCLK
PIN  2            /MEM_ACCESS ;
PIN  3            /REF_ACCESS ;
PIN  4            DELAY_IN ;
PIN  5            /WE0
PIN  6            /WE1
PIN  7            /WE2 ;
PIN  8            /WE3 ;
PIN  9            /WRITE
PIN  10           GND
PIN  11           /ST1 ;
PIN  13           /CAS0
PIN  14           /CAS1
PIN  15           /CAS2 ;
PIN  16           /CAS3 ;
PIN  17           /MUX ;
PIN  18           /RAS0
PIN  20           VCC ;

;--------------- Boolean Equation Segment ---------------


EQUATIONS


MINIMIZE_OFF

RAS0 =    MEM_ACCESS
        + REF_ACCESS*/MEMCLK
        + RAS0*REF_ACCESS


MUX =     MEM_ACCESS*/MEMCLK
        + MUX*MEM_ACCESS


CAS0 =    REF_ACCESS
        + MUX*MEM_ACCESS*ST1*/WRITE*MEMCLK*DELAY_IN
        + MUX*MEM_ACCESS*ST1*/WRITE*CAS0*DELAY_IN
        + MUX*MEM_ACCESS*ST1*/WRITE*CAS0*/MEMCLK
        + MUX*MEM_ACCESS*ST1*WRITE*WE0*/MEMCLK
```

```
CAS1 =    REF_ACCESS
        + MUX*MEM_ACCESS*ST1*/WRITE*MEMCLK*DELAY_IN
        + MUX*MEM_ACCESS*ST1*/WRITE*CAS1*DELAY_IN
        + MUX*MEM_ACCESS*ST1*/WRITE*CAS1*/MEMCLK
        + MUX*MEM_ACCESS*ST1*WRITE*WE1*/MEMCLK

CAS2 =  REF_ACCESS
        + MUX*MEM_ACCESS*ST1*/WRITE*MEMCLK*DELAY_IN
        + MUX*MEM_ACCESS*ST1*/WRITE*CAS2*DELAY_IN
        + MUX*MEM_ACCESS*ST1*/WRITE*CAS2*/MEMCLK
        + MUX*MEM_ACCESS*ST1*WRITE*WE2*/MEMCLK

CAS3 =    REF_ACCESS
        + MUX*MEM_ACCESS*ST1*/WRITE*MEMCLK*DELAY_IN
        + MUX*MEM_ACCESS*ST1*/WRITE*CAS3*DELAY_IN
        + MUX*MEM_ACCESS*ST1*/WRITE*CAS3*/MEMCLK
        + MUX*MEM_ACCESS*ST1*WRITE*WE3*/MEMCLK

;

;PALASM Design Description

;--------------- Declaration Segment ---------------
TITLE      MEMORY CYCLE GENERATOR FOR EZ-030
PATTERN    MSTR_CON.PDS
REVISION   A
COMPANY    AMD
DATE       03/03/92

CHIP  U4  PALCE16V8
; correction 4/25/92
; CS which is used for read-narrow initialization of the Am29030 at reset
; was set to 0 should be 1 correction below
;
;--------------- PIN Declarations ---------------
PIN 1           MEMCLK              ;
PIN 2           /RESET              ;
PIN 3           /REQ
PIN 4           A31
PIN 5           A30
PIN 6           /BURST              ;
PIN 7           /EMUL               ;
PIN 8           /SERIAL_RDY         ;
PIN 9           /REF_REQ            ;
PIN 10          GND                 ;
PIN 11          /OE                 ;
PIN 12          /RDY
PIN 13          /MEM_ACCESS         REGISTERED ;
PIN 14          /REF_ACCESS         REGISTERED ;
PIN 15          /ROM_CS
PIN 16          /IDLE               REGISTERED ;
PIN 18          /MEM_RDY            ;
PIN 19          /ST1                ;
PIN 20          VCC                 ;
```

```
;--------------- Boolean Equation Segment ---------------
EQUATIONS

IDLE :=    /IDLE*/MEM_ACCESS*/REF_ACCESS
         + RESET
         + /IDLE*MEM_ACCESS*ST1*MEM_RDY*/BURST
         + /IDLE*REF_ACCESS*/REF_REQ
         + IDLE*/(REQ*A31*/A30)*/REF_REQ*/EMUL


REF_ACCESS :=    IDLE*REF_REQ*/REF_ACCESS
               + REF_ACCESS*REF_REQ


MEM_ACCESS :=    IDLE*REQ*A31*/A30*/REF_REQ*/MEM_ACCESS*EMUL
               + MEM_ACCESS*/ST1
               + MEM_ACCESS*BURST


ST1 :=     MEM_ACCESS*/MEM_RDY
         + ST1*BURST


MEM_RDY :=       REQ*/A31*/A30*/MEM_RDY
               + MEM_ACCESS*/MEM_RDY
               + MEM_ACCESS*MEM_RDY*BURST


ROM_CS = /EMUL*REQ*/A31*/A30


RDY = MEM_RDY + SERIAL_RDY


;


;PALASM Design Description

;--------------- Declaration Segment ---------------
TITLE      REFRESH TIMER FOR EZ-030
PATTERN    REFRESH.PDS
REVISION   A
COMPANY    AMD
DATE       03/03/92

CHIP  U2  PALCE16V8

;----------------  PIN Declarations ---------------
PIN  1          MEMCLK                   ;
PIN  2          /REF_ACCESS              ;
PIN  11         /OE                      ;
PIN  12         /REF_REQ      REGISTERED ;
PIN  13         /Q0           REGISTERED ;
PIN  14         /Q1           REGISTERED ;
PIN  15         /Q2           REGISTERED ;
PIN  16         /Q3           REGISTERED ;
PIN  17         /Q4           REGISTERED ;
PIN  18         /Q5           REGISTERED ;
PIN  19         /Q6           REGISTERED ;
```

```
;--------------- Boolean Equation Segment ---------------

EQUATIONS

Q0 := /Q0

Q1 := Q1 :+: Q0

Q2 := Q2 :+: (Q1*Q0)

Q3 := Q3 :+: (Q2*Q1*Q0)

Q4 := Q4 :+: (Q3*Q2*Q1*Q0)

Q5 := Q5 :+: (Q4*Q3*Q2*Q1*Q0)

Q6 := Q6 :+: (Q5*Q4*Q3*Q2*Q1*Q0)

REF_REQ :=   Q6*Q5*Q4*Q3*Q2*Q1*Q0
           + REF_REQ*/REF_ACCESS

;

TITLE      85C30 CONTROLLER
PATTERN    SERIAL.PDS
REVISION   A
COMPANY    AMD
DATE       04/20/1992

CHIP U18 PAL16V8

CLK /RESET /REQ A31 A30 /WRITE /INT  NC NC GND

/OE /8530_WR /8530_RD /8530_CS /NC1 /NC2 /NC3 /INTR*(0) /SERIAL_RDY     VCC

EQUATIONS

8530_CS = REQ*A31*A30

8530_RD :=   8530_CS*/WRITE*/8530_RD*/SERIAL_RDY
           + 8530_RD*/SERIAL_RDY + RESET

8530_WR :=   8530_CS*WRITE*/8530_WR*/SERIAL_RDY
           + 8530_WR*/NC1*/SERIAL_RDY
           + 8530_WR*/NC2*/SERIAL_RDY + RESET

NC1 := (8530_RD + 8530_WR)*/NC1*/SERIAL_RDY

NC2 := (8530_RD + 8530_WR)*(NC1:+:NC2)*/SERIAL_RDY

SERIAL_RDY := NC1*NC2*/SERIAL_RDY

INTR*(0) = INT
```

```
;PALASM Design Description

;------------------------------ Declaration Segment ---------------------------

TITLE      ADDRESS COUNTER AND INCREMENTER FOR THE EZ-030 SPEED FIX
PATTERN    ADD_CNTR.PDS
REVISION   A
AUTHOR     DAVID STOENNER
COMPANY    AMD
DATE       10/19/93

CHIP     U10      PALCE20V8


;------------------------------- PIN Declarations -----------------------------

PIN   1    MEMCLK
PIN   2    /INC
PIN   3    A2
PIN   4    A3
PIN   5    A4
PIN   6    A5
PIN   7    A6
PIN   8    A7
PIN   9    A8
PIN   10   A9
PIN   12   GND
PIN   13   /OE
PIN   14   /LOAD
PIN   22   Q0                         REGISTERED ;
PIN   21   Q1                         REGISTERED ;
PIN   20   Q2                         REGISTERED ;
PIN   19   Q3                         REGISTERED ;
PIN   18   Q4                         REGISTERED ;
PIN   17   Q5                         REGISTERED ;
PIN   16   Q6                         REGISTERED ;
PIN   15   Q7
PIN   24   VCC

;---------------------------- Boolean Equation Segment --------------------------


EQUATIONS


Q0 :=     INC*/Q0
        + LOAD*A2

Q1 :=     INC*(Q1 :+: Q0)
        + LOAD*A3

Q2 :=     INC*(Q2 :+: (Q1*Q0))
        + LOAD*A4
```

```
Q3 :=     INC*(Q3 :+: (Q2*Q1*Q0))
        + LOAD*A5

Q4 :=     INC*(Q4 :+: (Q3*Q2*Q1*Q0))
        + LOAD*A6

Q5 :=     INC*(Q5 :+: (Q4*Q3*Q2*Q1*Q0))
        + LOAD*A7

Q6 :=     INC*(Q6 :+: (Q5*Q4*Q3*Q2*Q1*Q0))
        + LOAD*A8

Q7 :=     INC*(Q7 :+: (Q6*Q5*Q4*Q3*Q2*Q1*Q0))
        + LOAD*A9

;---------------------------------------------------------------------------------
```