# PCMCIA Socket Interface Controller for the Am29200™ Microcontroller Family

**Advanced Micro Devices**

***Application Note***

*by Mark McClain and David Stoenner*

*This application note defines a simple, small, low-cost hardware interface to a PCMCIA PC Card socket from an Am29200 microcontroller family member.*

## OVERVIEW

This application note presents the hardware implementation of an interface between an Advanced Micro Devices Am29200 microcontroller family member and a Personal Computer Memory Card International Association (PCMCIA) PC Card Standard (Release 2.0) socket. This design is intended for use with any of the following microcontrollers: Am29200, Am29205™, Am29240™, Am29243™, Am29245™, and future members of the Am29200 microcontroller family that share signal or pinout compatibility with these processors.

One card socket is supported. The required integrated circuit hardware consists of one 8-bit register, three 8-bit buffers, two 8-bit transceivers, and control logic implemented by one MACH®215 Programmable Logic Device (PLD).

All of the signal definitions and design discussions in this application note are with reference to the September 1991 PCMCIA PC Card Standard Release 2.0. A copy of this specification may be obtained from the PCMCIA located at 1030B East Duane Ave. Suite G, Sunnyvale CA 94086, (408) 720–0107.

## FUNCTIONAL DESCRIPTION

### Schematics

As shown in the two schematic pages in Appendix A, the PC Card socket is isolated from the processor system by three 74LS244 address buffers (U1, U2, and U3) and two 74LS245 data transceivers (U6 and U7), so that a card may be inserted or removed during system operation without disturbing the system address and data buses.

The PC card address space is divided into a series of 2-Mbyte pages. Pages are selected by the upper address bits loaded into the 74LS374 page address register (U5).

The PCMCIA PC Card socket is shown as J1.

The +12-V Vpp voltage pins of the socket may be connected to the +12-V supply via a discrete pass-transistor switch provided by Q1 and Q2. The diode D1 holds the Q1 emitter one diode voltage drop above ground so that in order to turn on, the base of Q1 must be driven by a voltage greater than the sum of the Q1 base emitter voltage drop, plus the diode voltage drop. This keeps the turn-on voltage for Q1 well above the voltage level for a TTL Low, which ensures Q1 stays off until a TTL High level is driven on the Vpp_EN signal. Whenever Q2 is off, the Vpp pins are held at the Vcc voltage through resistor R10 as required by the PC Card specification.

The control logic is provided by a MACH215 PLD (U4) device, which integrates the equivalent of four 22RA8 Programmable Array Logic, CMOS Electrically-erasable (PAL®CE) devices into a single 44-pin PLCC package.

### Control Logic

The PC Card socket control logic implemented by the MACH215 device contains a control register, used by the processor to manage the different modes of operation, and a status register, where the current state of the socket can be monitored. These two registers are selected by address lines A23–A21, as defined by the address map described on page 3. A read within the register address range returns the status register contents on the I/D7–I/D0 bus lines, and a write loads all bits of the control register from the I/D7–I/D0 bus lines. Thus the status register is read-only and the control register is write-only. Software must maintain a copy of the control register if its current state must be known in order to modify some control bits without changing others.

#### Control Register

The control register bits are assigned as follows:

- Control[7]: PC Card enable
- Control[6]: Not used
- Control[5]: Vpp (+12 V) enable
- Control[4]: Disable wait-state generator
- Control[3]: Status change ($\overline{\text{STSCHG}}$) interrupt enable
- Control[2]: 0=16-bit wide card mode; 1=8-bit wide card mode
- Control[1]: PC Card reset
- Control[0]: 0=Memory Card mode; 1=I/O Card mode

Publication# **18498** Rev. **A** Amendment **/0**
Issue Date: **May 1994**

All Control bits are active High (1=active).

Control[7] is used to: output enable the PC Card socket address buffers; enable or clear the $\overline{\text{IOIS16}}$-signal latch; and select between the generation of Card_Inserted or Card_Extracted interrupts, depending on the respective disabled or enabled state of the PC Card socket.

Control[5] connects the +12-V supply to the Vpp1 and Vpp2 PC Card socket lines via a discrete pass-transistor switch. When these Vpp signals are not connected to +12 V, a resistor holds the lines at the Vcc voltage supply level as required by the PC Card specification.

Control[4] disables a wait-state generator that is used to guarantee a minimum cycle length. This wait-state generator is used for applications using the PC Card interface in Am29200 microcontroller family ROM space (see "Timing" on page 5 for more details).

Control[3] is used to enable the status change interrupt, which allows an interrupt request when the $\overline{\text{STSCHG}}$ signal goes active. This bit should be active only after an I/O-type PC Card has its I/O mode enabled. The $\overline{\text{STSCHG}}$ signal is a dual-function signal, which also serves as a battery voltage detect signal (BVD1) when the PC Card functions as a memory card. Thus, enabling the status change interrupt when the card is acting as a memory could produce a false interrupt.

Control[2] indicates 8-bit card mode when active. This disables the odd-byte card enable CE2 so that only CE1 is asserted during an access. This forces all accesses to be byte wide and conducted only on PC Card data lines D7–D0.

Control[1] forces the PC Card reset line active when set High.

Control[0] enables the $\overline{\text{IORD}}$ and $\overline{\text{IOWR}}$ output signals from the control logic when set High.

**Status Register**

The status register bits are assigned as follows:

- Status[7]: Card inserted (both $\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ asserted)
- Status[6]: PC Card socket write-protect pin
- Status[5]: PC Card socket RDY/IREQ pin
- Status[4]: Card_Extracted interrupt request
- Status[3]: Card_Inserted interrupt request
- Status[2]: PC Card socket $\overline{\text{IOIS16}}$-signal latch (I/O card mode only)
- Status[1]: PC Card socket BVD2, $\overline{\text{SPKR}}$ pin (I/O mode Speaker)
- Status[0]: PC Card socket BVD1, $\overline{\text{STSCHG}}$ pin (I/O mode Status Change)

The status register displays the current state of several PC Card signal pins and can be used to poll the state of the card or to identify the source of an interrupt request.

Status[7] is the combinatorial AND of the Card Detect 1 and 2 signals. This bit is High when both Card Detect signals are Low (asserted). Since this signal is not synchronized to the processor clock nor debounced, this bit should be read multiple times when checking the Card Inserted state so as to provide a software debounce of the signal.

Status[6] is the current state of the PC Card Write Protect (WP) signal. When this bit is High, the card is write protected and the PC Card Write Enable ($\overline{\text{WE}}$) signal is held inactive.

Status[5] is the inverted current state of the PC card socket Ready/Not-Busy (RDY/BSY) or I/O Interrupt ($\overline{\text{IREQ}}$) dual-function signal. When this bit is High, the card is busy or an I/O interrupt is being requested.

State[4] is the latched state of an interrupt request generated by either $\overline{\text{CD1}}$ or $\overline{\text{CD2}}$ going inactive while the card is in use with Control[7] active. When this bit is High, the Card_Extracted interrupt is being requested. This bit is cleared when Control[7] is inactive.

State[3] is the latched state of an interrupt request generated by both $\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ going active while the card is not in use with Control[7] inactive. When this bit is High, the Card_Inserted interrupt is being requested. This bit is cleared when Control[7] is active.

Status[2] is the latched indication that the $\overline{\text{IOIS16}}$-signal was active at any time while the card was in I/O mode with Control[0] set High.

Status[1] is the current state of the Battery Voltage Detect 2 (BVD2) signal from the PC Card socket. This is a dual-function signal. While a card is serving as a memory-only card, the signal indicates the state of the card battery. When a card is in I/O mode, the signal serves as a digital speaker output signal ($\overline{\text{SPKR}}$).

Status[0] is the current state of the Battery Voltage Detect 1 (BVD1) signal from the PC Card socket. This is a dual-function signal. While a card is serving as a memory-only card, the signal indicates the state of the card battery. When a card is in I/O mode, the signal serves as a status change indication. When Control[3] is set High, this signal is enabled as an interrupt request.

## Control Output Signals

The PC Card socket Reset signal (P_RESET) is the logical OR of the system reset signal and Control[1]. The system reset provides a hardware reset and Control[1] provides a software-controlled reset.

The address enable ($\overline{\text{ADD\_EN}}$) and Vpp enable (VPP_EN) signals are the direct outputs of the Control[7] and Control[5] register bits, respectively. The output enable for the I/O read ($\overline{\text{P\_IORD}}$) and I/O write ($\overline{\text{P\_IOWR}}$) signals is the Control[0] register bit.

The processor wait signal ($\overline{\text{PROC\_WAIT}}$) is the logical OR of the PC Card socket $\overline{\text{WAIT}}$ signal and the output of a wait-state generator in the control logic. In order to use some of the slower memories defined in the PC Card specification, a wait-state generator is required when this PC Card socket is used in the ROM space of an Am29200 microcontroller family member. This output is clocked by MEMCLK, since the $\overline{\text{WAIT}}$ input of these microcontrollers is a MEMCLK synchronous signal.

The Card Enable (CE1 and CE2) signals follow the chip select input to the control logic when the active address is within the common memory, attribute memory, or I/O regions of the address map during 16-bit accesses. However, CE2 is disabled if Control[2] indicates that the card is 8-bit only, or if the address is within the attribute memory region. When the active address is within the common-memory or attribute-memory regions, the PC Card Output Enable ($\overline{\text{P\_OE}}$) signal follows the processor output enable signal. The write enable ($\overline{\text{P\_WE}}$) follows the processor write enable if the PC Card Write Protect (P_WP) signal is inactive (LOW). The PC Card I/O read ($\overline{\text{P\_IORD}}$) and write ($\overline{\text{P\_IOWR}}$) signals follow the processor output enable and write enable signals when the active address is in the I/O region. The PC Card Register ($\overline{\text{P\_REG}}$) signal follows the chip select for accesses within the attribute or I/O address regions.

The Data Buffer Enable ($\overline{\text{DATA\_EN}}$) signal follows chip select and output enable or write enable for addresses within the common-memory, attribute-memory, or I/O regions. The Data Direction (DATA_DIR) signal follows the processor output enable.

The Interrupt Request ($\overline{\text{INTR}}$) signal is active for any of the following situations: there is a card inserted or card extracted interrupt request; the PC Card Interrupt Request signal is active and Control[0] is High (I/O card mode); or the PC Card Status Change ($\overline{\text{STSCHG}}$) signal is active and Control[3] is High (status change interrupt is enabled).

## Address Map

Accesses to the two 64-Mbyte address spaces of the PC Card socket and the implemented PCMCIA controller function are mapped within a 16-Mbyte window of the processor address space. Accesses are mapped as shown in the table below.

**Table 1. 29K™ Family 16-Mbyte Address Map**

| Address Range Plus Base Address (hex) | Address Lines A23–A21 | Function |
|---|---|---|
| 000000 thru 1FFFFF | 000 | PCMCIA Common Memory in a 8-bit or 16-bit card as determined by control register |
| 200000 thru 7FFFFF | 0xx | Reserved (accessed via the page address register) |
| 800000 thru 9FFFFF | 100 | PCMCIA Attribute Memory as 8-bit values on even bytes |
| A00000 thru BFFFFF | 101 | PCMCIA I/O access |
| C00000 thru DFFFFF | 110 | Read access to status register and write access to control register |
| E00000 thru FFFFFF | 111 | Write access to page address register; read access is ignored and the data bus floats |

This address map creates a 2-Mbyte window or page within each of the three addressable sections of a PC Card (the common-memory, attribute-memory, and I/O regions). Each of these regions has a potential address space of 64 Mbyte. A particular 2-Mbyte page within the 64-Mbyte address space is selected by a single-page address register (U5, 74LS374), which supplies address bits A25–A21 to the PC Card socket. The same page address is thus used by all three regions. This page address register is write-only so software must maintain a copy of the register contents, if the software needs access to the current value in the register.

## INTERFACE SELECTION

The 16-Mbyte address window for the PC Card interface can be placed in a ROM bank or connected to a Peripheral Interface Adapter (PIA) port. Table 2 summarizes the interface features and support by ROM and PIA for these features. These features and support are described in more detail following Table 2.

**Table 2. Interface Option Summary**

| Feature | Memory Space | |
|---|---|---|
| | ROM | PIA |
| XIP | Supported | Not supported |
| 8- vs. 16-bit cards | Restricted by system design to either 8- or 16-bit cards | Both 8- and 16-bit cards supported |
| Automatic word assembly and splitting | Yes | No |
| Little-endian to big-endian order conversion | Must support via software | Must support via software |
| Unaligned accesses | Support via software | Support via software |
| DMA support | No | Yes |
| System clock rate | MEM CARD up to 20 MHz; I/O Read up to 14 MHz; I/O Write up to 7 MHz | Memory and I/O supported up to 20 MHz |

## ROM Bank

Placement in a ROM bank provides the following:

- Execute-In-Place (XIP) support for PC Cards containing program code
- Automatic assembly of 32-bit word and 16-bit half-word values read from 16-bit or 8-bit wide PC Cards
- Automatic splitting of 32-bit values when written into 16-bit wide PC Cards (the Am29200 microcontroller family does not support writing into 8-bit wide devices in ROM space; this must be done via a software read-modify-write on a 16-bit value)

However, placement in a ROM bank *requires* the following:

- The Large Memory bit of the ROM control register must be set active to enable 16-Mbyte maximum bank size, which limits the minimum memory bank size within any ROM bank to 2 Mbyte. This requirement might need to be considered if all ROM banks must be arranged to form a single contiguous address space (refer to the ROM controller address mapping section of the *Am29200 and Am29205 RISC Microcontrollers User's Manual*, order# 16362, for additional information).
- The code and data values accessed as quantities wider than that supported by the PC Card (e.g., a 32-bit word read from a 16-bit wide card) must be placed in Big-Endian order (most-significant byte in lowest address position). Since the Am29200 microcontroller family supports only Big-Endian ordering, any Little-Endian ordered data in the PC Card must be supported by software that converts between Big-Endian order in the Am29200 microcontroller family system memory, and Little-Endian order in the PC Card.
- Accesses of half-words or words not aligned to half-word or word boundaries, respectively, are not supported by processor hardware and must be performed by software.
- The data bus of a 16-bit card must be connected to I/D31–I/D16 and the data bus of an 8-bit card must be connected to I/D31–I/D24. This implies one of the following: that only one type of card can be supported by prewiring the data bus, that an extra data transceiver is needed to connect the lower half of the 16-bit PC Card data bus to the upper byte of the I/D bus when an 8-bit card is in use, or that 8-bit card accesses to I/D23–I/D16 must be done only by software (eliminating XIP support).
- DMA to or from the PC Card cannot be supported by Am29200 or Am29205 microcontrollers, although the Am29240 microcontroller series can support DMA in ROM space.
- Some hold-time parameters, defined for one type of memory PC Card, will be violated, which might cause incompatibility with some PC Cards. Also, I/O type cards are usable only in systems operating at lower than normal clock rates (these issues are described in more detail in the Timing section on page 5).
- Execute-in-place (XIP) code size must reside within a 2-Mbyte page because of the page register. If code size larger than 2 Mbyte is needed, then the code must manipulate the page register.

## PIA Port

Connection to a PIA port provides the following:

- Processor built-in wait-state generator support for accesses as slow as 32 system clock cycles, without the need to use the PC Card $\overline{\text{WAIT}}$ signal
- Support for DMA accesses to or from a PC Card by any Am29200 microcontroller family member

However, a PIA port *requires* the following:

- No code execution/execute-in-place can be supported since the Am29200 microcontroller family is unable to fetch code from the PIA regions.
- Data values wider than the PC Card width (e.g., accessing a 32-bit word in a 16-bit wide PC Card) must be assembled during reads or split during writes explicitly by software since there is no bus interface hardware support for these functions in the PIA regions. This also means Little-Endian data byte order in the PC Card must be handled by the software that accesses the PC Card.

- The 16-bit PC Card data bus must be connected to I/D15–I/D0.
- The I/O Extend bit for the PIA region used must be set High in the PIA Control register in order to provide the correct hold times relative to PIAOE and PIAWE.

When placed in a ROM bank, the processor's ROM Chip Select ($\overline{\text{ROMCS}}$) for the bank used is connected to the chip select input of the control logic. The ROM Output Enable ($\overline{\text{ROMOE}}$) and Write Enable ($\overline{\text{RSWE}}$) from the processor are connected to the control logic output enable and write enable inputs.

When connected to a PIA port, the PIA Chip Select ($\overline{\text{PIACS}}$) from the processor for the port used is connected to the chip select input of the control logic. The PIA Output Enable ($\overline{\text{PIAOE}}$) and Write Enable ($\overline{\text{PIAWE}}$) from the processor are connected to the control logic output enable and write enable inputs.

## TIMING

## ROM Bank

The Am29200 microcontroller family has its own separate ROM space wait-state generator but is limited to producing a maximum 4-cycle long access (250 ns at 16 MHz). That access length is too short for use with Attribute Memory, specified with a minimum access time of 300 ns.

This is why the MACH215 device control logic includes a simple wait-state generator to drive the $\overline{\text{WAIT}}$ line active and extend accesses for use with the slower memories defined in the PC Card standard.

The wait-state generator drives the processor $\overline{\text{WAIT}}$ line for four cycles, beginning with the assertion of either $\overline{\text{ROMOE}}$ or $\overline{\text{RSWE}}$. Following the fifth cycle, in which the processor $\overline{\text{WAIT}}$ line is deasserted, the processor takes one additional cycle to complete the access. Since the $\overline{\text{ROMOE}}$ is not asserted until the beginning of the second cycle of an access, the total length of a read access will be seven cycles. This provides a minimum 350-ns access period, with a system clock at 20 MHz. During a write access, $\overline{\text{RSWE}}$ is asserted during the first cycle of the access, therefore write accesses will be six cycles long. Control[4] bit is cleared during reset so that the wait-state generator is enabled, allowing Attribute Memory to be accessed after reset.

The Am29200 microcontroller family ROM control register for the bank used must also be programmed for two or more wait states. This allows the wait signal, from the PC Card control logic, to be asserted before the normal end of access with enough setup time to affect an extension of the access. If the card contains faster memory (or PIA space is used), which needs fewer wait states, the control logic wait-state generator can be disabled by setting Control[4], and the microcontroller wait-state generator can be set for fewer wait states.

In either case, the Wait signal from the PC Card interface is logically OR'ed into the processor $\overline{\text{WAIT}}$ line so that the PC Card can extend accesses as needed.

Note that 600-ns access time common memories are not supported since those cards are restricted to 3.3-V operation and this interface has been designed only for 5-V operation.

The key timing parameters are described below.

### Memory Read

There is a required 30-ns address setup time to Output Enable. The Am29200 microcontroller family Output Enable goes active one cycle after Address and ROMCS go active, providing 1 clock cycle of Address Setup time. This is sufficient for systems operating at 20 MHz using 300-ns access time PC Cards.

There are 20-ns hold times defined from Output Enable going inactive until Address or Card Enable can go inactive. These are more difficult specifications since the Am29200 microcontroller family ROM interface disables Address, ROM Chip Select, and Output Enable at the same clock edge—at the end of a read access.

Meeting these specifications requires that the PCMCIA Output Enable ($\overline{\text{P\_OE}}$) be made inactive before the end of the ROM access. This could be done by deasserting the $\overline{\text{WAIT}}$ signal at the same time that $\overline{\text{P\_OE}}$ is forced inactive. The processor would then take one additional cycle to end the access after $\overline{\text{WAIT}}$ is deasserted. This would provide one cycle of Address and Card Enable hold time as well as additional Output Disable time, which would allow operation in 20-MHz systems with 300-ns PC Cards. However, Output Enable ($\overline{\text{P\_OE}}$) to the PC Card cannot be disabled prior to the end of the access without also losing the data being read, unless the data bus transceiver is replaced by a latching data transceiver so as to hold the data valid after Output Enable is deasserted.

Note that this suggested Output Enable timing and latching data transceiver are *not* implemented in this design. The PC Card Output Enable and Card Enable hold times are thus violated. This might cause some PC Cards to not function properly with this design. This choice was made because: these specifications are only defined for cards supporting the $\overline{\text{WAIT}}$ signal, thus many PC Cards would not rely on these specifications and should function without them; the specifications seem to serve no purpose and are not explained by the PC Card standard; and the additional complexity and cost involved can be avoided by accepting a minimal risk of incompatibility with real PC Cards.

### Memory Write

During a write operation, the Am29200 microcontroller family asserts the Write Enable ($\overline{\text{RSWE}}$) one half cycle after the beginning of an access and deasserts the sig-

nal one half cycle before the end of an access. This provides one half cycle for Address Setup to Write Enable, Data Hold, and Write Recovery time. This allows system operation up to 16 MHz with 250-ns memories and 20 MHz with 200-ns memories.

#### I/O Read

There is a required address setup time to I/O Read ($\overline{IORD}$) of 70 ns. The Am29200 microcontroller family Output Enable is used to create $\overline{IORD}$. Output Enable goes active one cycle after Address and ROM select go active, providing one clock cycle of address setup time. This means that I/O accesses are restricted to systems operating below 14.2 MHz.

As with the memory read operation, there are Card Enable and Address-to-IORD hold times defined. As explained earlier, the Am29200 microcontroller family ROM interface cannot directly support this hold-time requirement. These specifications are also violated by this design.

#### I/O Write

I/O Write ($\overline{IOWR}$) is derived from the Am29200 microcontroller family Write Enable. During a write operation, the microcontroller asserts the Write Enable ($\overline{RSWE}$) one-half cycle after the beginning of an access and deasserts the signal one-half cycle before the end of an access. This provides one-half cycle relative to Write Enable for Address, Data, and Card Enable Setup, and Data, Address, and Card Enable Hold. The largest of these parameters is 70 ns, thus allowing system operation during I/O writes only below 7.1 MHz.

### PIA Port

The Am29200 microcontroller family has a more flexible built-in wait-state generator for the PIA ports. The PIA port wait-state generator can extend accesses up to 31 wait states, in addition to the one cycle required for any access. So, without using the Am29200 microcontroller family $\overline{WAIT}$ input to further extend an access, the maximum length access is 32 cycles. Also the Output Enable and Write Enable signals are asserted during the second cycle of an access, thus providing additional address setup time. When the I/O Extend (IOEXTx) bit of the microcontroller PIA Control register is set, the end of a PIA access is extended by one additional cycle after Output Enable ($\overline{PIAOE}$) goes inactive on a read, or by two cycles after Write Enable ($\overline{PIAWE}$) goes inactive on a write. This allows for additional Address and Card Enable hold, and Output Disable time needed by the PC Card standard. However, note that 600-ns memories are not intended for use with this interface due to their 3.3-V operation. All other defined memory and I/O device timing can be supported directly by the PIA wait-state generator for system speeds up to 20 MHz.

### OTHER DESIGN OPTIONS

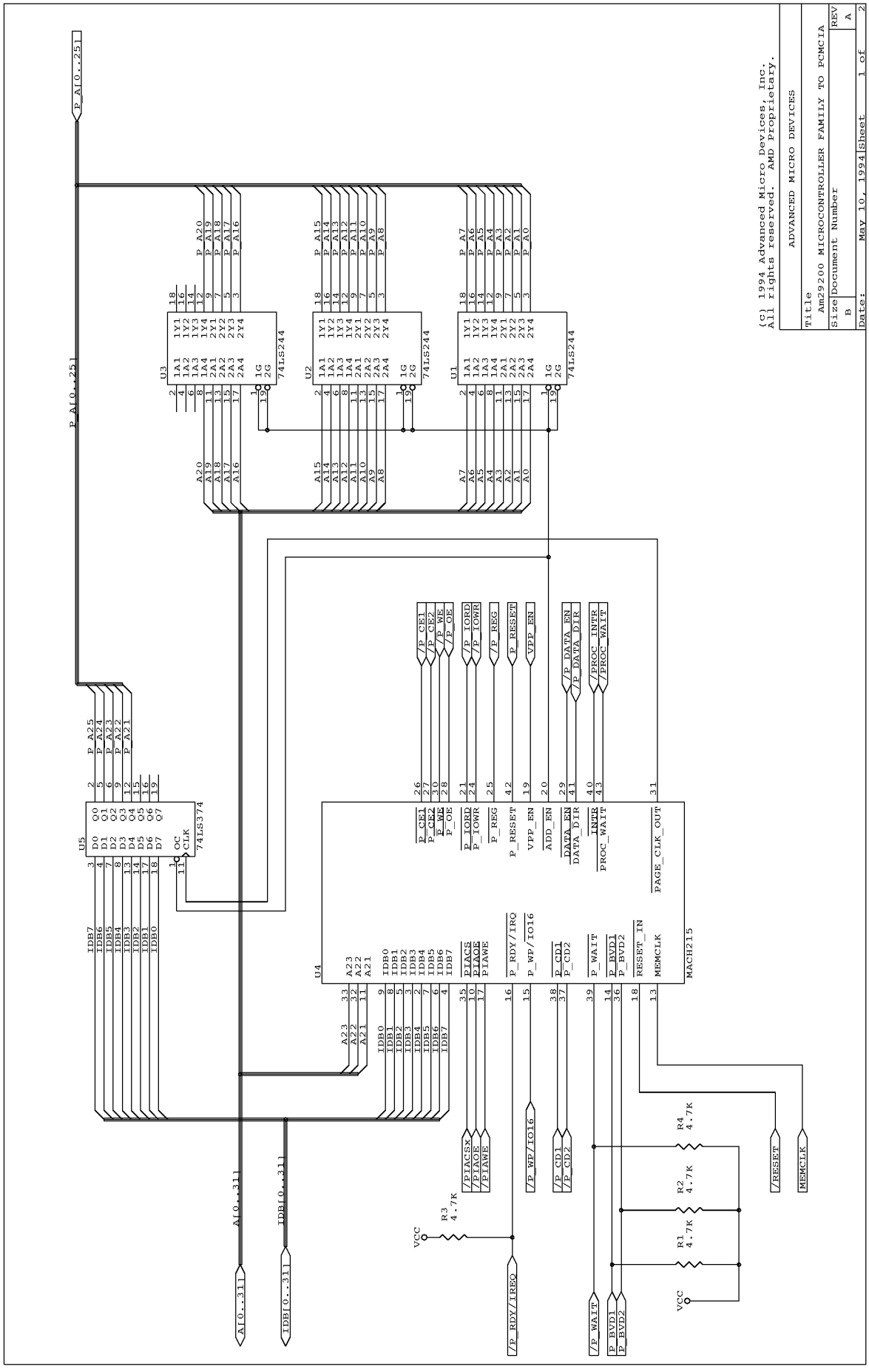
If the system in which this PC Card socket design is to be used has no requirement to insert or remove cards during system operation, then the address and data buffers can be eliminated to reduce system cost and board space needs.

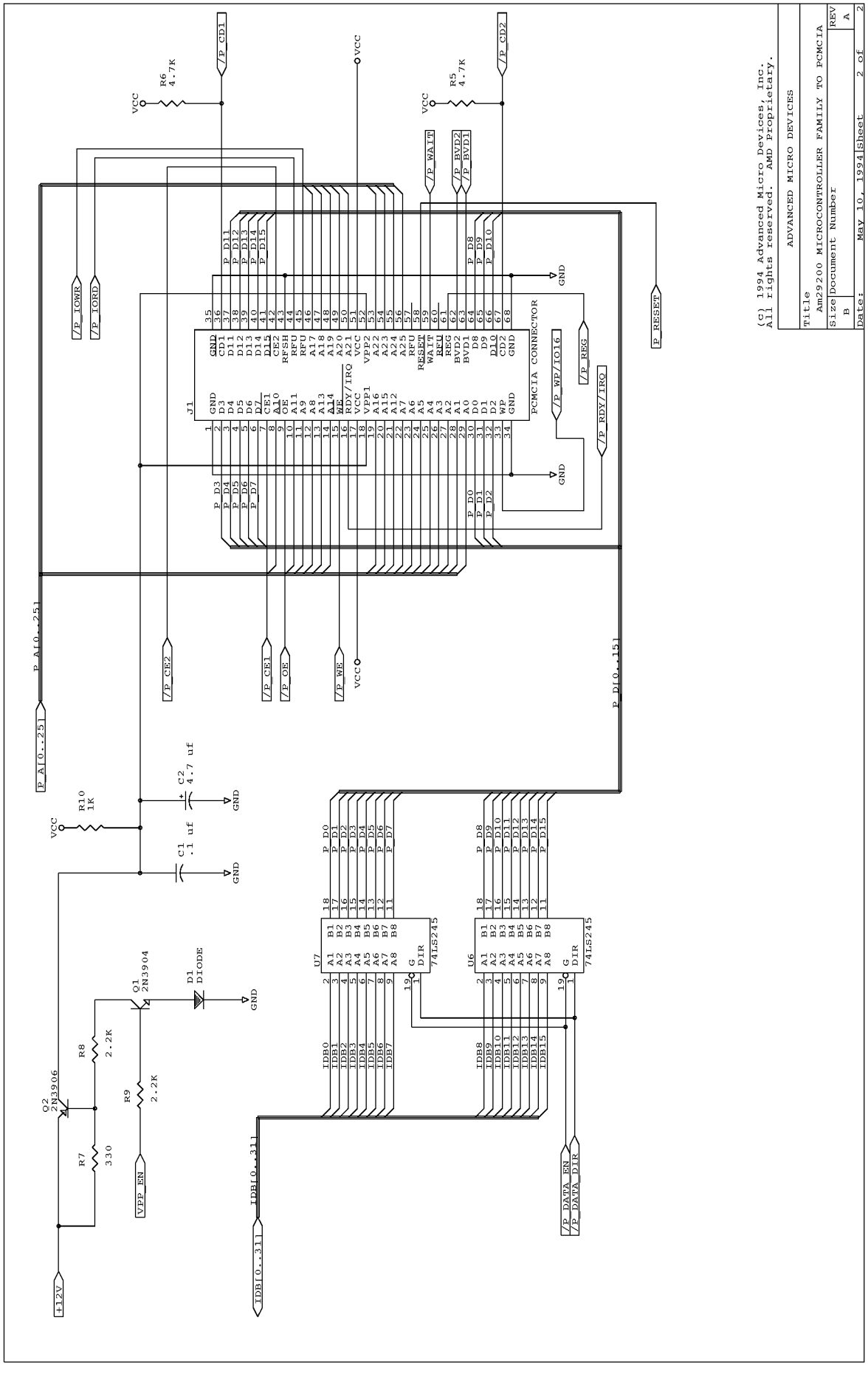
### CONCLUSION

This application note defines a simple, small, low-cost hardware interface to a PCMCIA PC Card socket. All control logic is contained in a single 44-pin PLCC MACH215 Programmable Logic Device. All the remaining components are low-cost, standard buffers and transceivers that are available in SOIC packages to minimize board space. There are no high performance requirements for the interface logic so lower speed (low-cost) components may be selected and the number of wait states set accordingly. As of the date on this publication, the cost for the MACH215 device (20-ns tpd) is $5.65 each (with a quantity of 100). The other logic devices are estimated at $0.50 each, for a total of about $3.00. Thus, the logic cost for this design is about $9.00. These prices may have changed however, so check with your AMD sales representative for the latest price.

## Appendix A. Schematics

The schematics for this design are shown on the pages that follow.


P_A[0..25]
A[0..31]
IDB[0..31]
U1
U2
U3
74LS244
U4
MACH215
U5
74LS374
R1 4.7K
R2 4.7K
R3 4.7K
R4 4.7K
VCC
/RESET
MEMCLK
/P_WAIT
P_BVD1
P_BVD2
/P_RDY/IRQ
/P_WP/IO16
/P_CD1
/P_CD2
/PIACS
/PIAOE
/PIAWE
/P_CE1
/P_CE2
/P_WR
/P_OE
/P_IORD
/P_IOWR
/P_REG
P_RESET
VPP_EN
ADD_EN
DATA_EN
DATA_DIR
/PROC_INTR
/PROC_WAIT
PAGE_CLK_OUT
RESET_IN
MEMCLK
(C) 1994 Advanced Micro Devices, Inc.
All rights reserved. AMD Proprietary.
ADVANCED MICRO DEVICES
Title
Am29200 MICROCONTROLLER FAMILY TO PCMCIA
Size B
Document Number
REV A
Date: May 10, 1994 Sheet 1 of 2


(C) 1994 Advanced Micro Devices, Inc.
All rights reserved. AMD Proprietary.
ADVANCED MICRO DEVICES
Title
Am29200 MICROCONTROLLER FAMILY TO PCMCIA
Size B
Document Number
REV A
Date: May 10, 1994 Sheet 2 of 2
PCMCIA CONNECTOR
74LS245
2N3904
2N3906
DIODE
P_A[0..25]
P_D[0..15]
IDB[0..31]
VCC
GND
+12V
/P_IOWR
/P_IORD
/P_CE1
/P_CE2
/P_OE
/P_WE
/P_WAIT
/P_BVD1
/P_BVD2
/P_RESET
/P_REG
/P_WP/IO16
/P_RDY/IRQ
/P_CD1
/P_CD2
/P_DATA_EN
/P_DATA_DIR
VPP_EN

## APPENDIX B. PAL Equations

This appendix shows the PAL equations in PALASM® software syntax for the MACH215 device PCMCIA PC Card Interface Control Logic Design.

```
;PALASM Software Design Description
;---------------------------------- Declaration Segment ------------
TITLE    PCMCIA CONTROLLER FOR Am29200 MICROCONTROLLER FAMILY PROCESSOR SYSTEMS
PATTERN  PCMCIA3.PDS
REVISION E
AUTHOR   DAVID STOENNER / MARK MC CLAIN
COMPANY  ADVANCED MICRO DEVICES
DATE     1/19/94

CHIP  U4  MACH215
;---------------------------------- PIN Declarations ---------------
PIN 13     MEMCLK ; Clock
PIN 11, 32, 33 A[21..23]   ; Input
PIN 18     /RESET_IN       ; Input
PIN 38     /P_CD1 ; Input
PIN 37     /P_CD2 ; Input
PIN 39     /P_WAIT         ; Input
PIN 35     /PIACS ; Input
PIN 10     /PIAOE ; Input
PIN 17     /PIAWE ; Input
           ; If this controller is used in the ROM space of the Am29200
           ; microcontroller family, then the PIACS should be replaced with the
           ; ROMCS of interest, the PIAOE should be replaced with ROMOE, and
           ; the PIAWE should be replaced with RSWE.  This will allow
           ; XIP of a memory card in ROM space.
PIN 16     /P_RDY_IRQ      ; Input
           ; This pin has a dual function as the RDY/BUSY pin for
           ; a memory card, and as the interrupt-request pin for an I/O
           ; card.
PIN 15     P_WP   ; Input
           ; This pin has a dual function as the write-protect pin for
           ; memory, and the /IO16 pin for an I/O card.
PIN 14     /P_BVD1         ; Input
PIN 36     /P_BVD2         ; Input
           ; These two pins have dual function as the battery voltage
           ; detect for a memory card, and as status change
           ; and speaker output, respectively, for an I/O card.

PIN 9, 8, 5, 3, 2, 7, 6, 4 IDB[0..7] ; Input/Output

PIN 26     /P_CE1 ; Output
PIN 27     /P_CE2 ; Output
PIN 30     /P_WE  ; Output
PIN 28     /P_OE  ; Output
PIN 21     /P_IORD         ; Output
PIN 24     /P_IOWR         ; Output
PIN 25     /P_REG ; Output
PIN 42     P_RESET         ; Output
PIN 19     VPP_EN ; Output
PIN 20     /ADD_EN         ; Output
PIN 29     /DATA_EN        ; Output
PIN 41     /DATA_DIR       ; Output
PIN 40     /INTR  ; Output
PIN 43     PROC_WAIT       ; Output
```

```
          ; This is the control for the processor to finish a
          ; transaction.  On the Am29200 microcontroller family, this
          ; is a negative true wait so this signal must be positive true in
          ; logic sense to finish a processor transaction.  For an Am29000® or
          ; Am29030™ processor, this is a negative true ready so simply put a /
          ; in front of PROC_WAI.

PIN 31     /PAGE_CLK_OUT    ; Output

NODE 32, 30, 28, 26, 24 CONTROL[0..4]
NODE 8     IO_CARD_16       OPAIR IDB[2]
NODE 4     CARD_INSERTED    OPAIR IDB[3]
NODE 2     CARD_EXTRACTED   OPAIR IDB[4]
NODE 58, 60 WT_ST[0..1]

;GROUP MACH_SEG_A IDB[0..7]
;GROUP MACH_SEG_B CONTROL[0..4] ADD_EN VPP_EN

STRING CON_CLK 'A[23]*A[22]*/A[21]*PIACS*PIAWE'

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

; The upper address bits A21 thru A23 are used to decode the PCMCIA
; access as follows:

;    0x000000 THRU 0x1FFFFF   PCMCIA Common Memory as 8-bit/16-bit
;                             card as determined by CONTROL2
;    0x200000 THRU 0x7FFFFF   Reserved
;                             (implemented through the page register)
;    0x800000 THRU 0x9FFFFF   PCMCIA Attribute Memory as 8 bit
;    0xA00000 THRU 0xBFFFFF   PCMCIA I/O access
;    0xC00000 THRU 0xDFFFFF   Read is STATUS and write is CONTROL
;    0xE00000 THRU 0xFFFFFF   Write to Address Page Register
;                             with D3 to D7 for P_A23 to P_A25;
;                             read is ignored and the data bus floats
;
; The Control Register is assigned as follows:
;
;    CONTROL7  PCMCIA enable
;    CONTROL6  Not used
;    CONTROL5  Turn on VPP (+12V) to card
;    CONTROL4  Disable wait-state generator
;    CONTROL3  Enable BVD1 (Status Change) interrupt
;    CONTROL2  8-bit memory card
;    CONTROL1  Reset PCMCIA interface
;    CONTROL0  0=memory card, 1=I/O card
;
; The Status Register is as follows:
;
;    STATUS7   Card Inserted (both CD1 and CD2 TRUE)
;    STATUS6   Write Protect pin
;    STATUS5   RDY/IRQ pin state
;    STATUS4   CARD_EXTRACTED, detected
;    STATUS3   CARD_INSERTED, detected
;    STATUS2   If card is in I/O mode and IO16 was true for at least
;              one access
;    STATUS1   BVD2 (Speaker for I/O card)
```

```
;    STATUS0   BVD1 (Status Change for I/O card)
;-------------------------------------------------------------------

PAGE_CLK_OUT = A[23]*A[22]*A[21]*PIACS*PIAWE

CONTROL[0..4].CLKF = CON_CLK

CONTROL[0..4] := IDB[0..4]

CONTROL[0..4].RSTF = RESET_IN

P_RESET = CONTROL[1] + RESET_IN

VPP_EN.CLKF = CON_CLK

VPP_EN := IDB[5]

ADD_EN.CLKF = CON_CLK

ADD_EN := IDB[7]

P_IORD.TRST = CONTROL[0]

P_IOWR.TRST = CONTROL[0]

P_IOWR = A[23]*/A[22]*A[21]*PIACS*PIAWE

P_IORD = A[23]*/A[22]*A[21]*PIACS*PIAOE

P_WE = PIACS*PIAWE*/P_WP*/(A[23]*/A[22]*A[21])

P_OE = PIACS*PIAOE*/(A[23]*/A[22]*A[21])

P_CE1 =   /A[23]*PIACS                  ; access as a 16-bit Common Mem
        + A[23]*/A[22]*/A[21]*PIACS     ; access for the Attribute Mem
        + A[23]*/A[22]*A[21]*PIACS      ; access as an I/O device

P_CE2 =   /A[23]*/CONTROL[2]*PIACS      ; access as a 16-bit full word
        + A[23]*/A[22]*A[21]*/CONTROL[2]*PIACS
                                        ; access as a 16-bit I/O device

P_REG =   A[23]*/A[22]*/A[21]*PIACS     ; access for Attribute Memory
        + A[23]*/A[22]*A[21]*PIACS      ; access for I/O device

DATA_EN = PIACS*((/A[23] + A[23]*/A[22])*(PIAOE + PIAWE))

DATA_DIR = PIAOE

PROC_WAIT.TRST = DATA_EN

PROC_WAIT.CLKF = MEMCLK

PROC_WAIT :=   /CONTROL[4]*DATA_EN*/P_WAIT*WT_ST[1]*WT_ST[0]
             + CONTROL[4]*DATA_EN*/P_WAIT

IO_CARD_16 =   ADD_EN*CONTROL[0]*/P_WP
             + ADD_EN*IO_CARD_16*CONTROL[0]
```

```
CARD_INSERTED =   /ADD_EN*P_CD1*P_CD2   ; is only generated when the unit is
                + /ADD_EN*CARD_INSERTED ; not enabled

CARD_EXTRACTED =   ADD_EN*/P_CD1*/P_CD2 ; is generated on an enabled card that
                                        ; is withdrawn
                 + ADD_EN*CARD_EXTRACTED

IDB[0..7].TRST = A[23]*A[22]*/A[21]*PIACS*PIAOE

IDB[7] = P_CD1*P_CD2

IDB[6] = P_WP

IDB[5] = P_RDY_IRQ

IDB[4] = { CARD_EXTRACTED }

IDB[3] = { CARD_INSERTED }

IDB[2] = { IO_CARD_16 }

IDB[1] = P_BVD2

IDB[0] = P_BVD1

INTR =    CARD_INSERTED
        + CARD_EXTRACTED
        + CONTROL[0]*P_RDY_IRQ          ; IRQ from I/O card is TRUE
        + CONTROL[3]*P_BVD1             ; if either the battery voltage
                                        ; or I/O card Status Change
                                        ; become TRUE

WT_ST[0..1].CLKF = MEMCLK

WT_ST[0..1].RSTF = RESET_IN

WT_ST[0] :=       DATA_EN*/WT_ST[0]         ;toggle LSB counter
                + DATA_EN*WT_ST[1]*WT_ST[0] ;hold at 11 until access end

WT_ST[1] :=       DATA_EN* (WT_ST[1] :+: WT_ST[0])  ;MSB count
                + DATA_EN*WT_ST[1]*WT_ST[0] ;hold at 11 until access end

;---------------------------------- END---------------------------
```