



Dear PSD3XX user:

We are pleased to announce the availability of the new **CEIBO** DS-300 PSD Emulator (directly from **CEIBO** in Florissant, Missouri). This powerful new tool has been designed to assist you during your design phase with programmable PSD3XX MCU peripherals.

The **CEIBO** DS-300 is loaded with features to help you significantly reduce system development time. The graphic configuration software allows you to configure the chip and, by using the on-board compiler, generate a file that can be downloaded to the emulator for emulation and debugging purposes. This file will ultimately be sent to your programmer to program the PSD3XX.

Because of the graphic environment and the ability to compile your file directly from these inputs, learning time is practically zero. Install the software, bring up the proper windows and go to work. It's that simple.

We urge you to take a couple of minutes to read the enclosed **CEIBO** DS-300 data sheet. We believe you will be impressed with the system's capabilities. The DS-300 will enable you to get your design to market quickly.

The DS-300 includes a 15-day evaluation period and a two-year parts and labor warranty so there is no risk involved. Give it a try, you have nothing to lose but excessive development time.

Please note the DS-300 is available only from **CEIBO** and is not available from WSI, Inc. On the back of this page is a **CEIBO** order form. **CEIBO** accepts the following payment terms:

- ◆ Net 30 with a purchase order and credit references.
- ◆ Visa or Mastercard.
- ◆ C.O.D.
- ◆ Prepayment via Bank transfer.

Thank you for your interest in the **CEIBO** DS-300 and the WSI PSD3XX.

The WSI PSD Marketing Department

# CEIBO DS-300 Order Form

Fax, mail or call  
 CEIBO USA  
 7 Edgestone Ct.  
 Florissant MO 63033  
 1-800-833-4084 or (314)830-4084 FAX (314)830-4083

<b>FROM:</b>  	<b>Ship to (if different address):</b>  
----------------------	--

P.O. number	PLACED BY	TERMS(see below)	SHIP VIA	F.O.B.	DATE EXPECTED
				ST. LOUIS, MO	

QTY.	DESCRIPTION	UNIT PRICE	TOTAL
	DS-300 PSD emulator for PSD3XX devices. Complete system including emulator with 44-pin PLCC header, PSDMAP configuration software, User's manual, power supply and RS-232 interface cable.	990.00	

	SHIPPING*	
	SUBTOTAL	
	SALES TAX RATE	
	SALES TAX**	
	<b>TOTAL DUE</b>	

Authorized Signature \_\_\_\_\_

NET 30 - ATTACH P.O. & CREDIT REFERENCES

VISA/MASTERCARD - NUMBER \_\_\_\_\_ EXP. DATE \_\_\_\_\_

C.O.D.

INTERNATIONAL ORDERS

PLEASE MAKE BANK TRANSFER THROUGH FED NET TO:

CEIBO INC  
 ABA NUMBER 081016932  
 MAGNA BANK OF MISSOURI  
 ACCOUNT NUMBER 3033300146

\* Shipping charges vary depending on type and destination.

\*\* Add sales tax of 6.25% if ordering within the state of MO.

Immeuble OSLO - 19, Avenue de Norvège  
Z.A. de Courtaboeuf - BP 3  
91 941 LES ULIS Cedex FRANCE  
Téléphone : 01 69 07 08 24  
Fax : 01 69 07 17 23  
[microel@wanado.fr](mailto:microel@wanado.fr)  
[www.microel.fr](http://www.microel.fr)

Les Ulis, le 11 janvier 1999

Madame, Monsieur,

Je vous prie de trouver ci-joint une mise à jour du logiciel de programmation des composants WSI. Cette mise à jour vous permet de programmer les nouvelles familles de PSD que WSI propose, à savoir les familles Easy FLASH<sup>TM</sup>. Ces nouvelles familles méritent toute votre attention : complètement en technologie Flash, elles sont reprogrammables à volonté par liaison JTAG. D'un coût similaire à une solution discrète, elles vous faciliteront la réalisation de vos cartes à base de microcontrôleur, réduisant considérablement l'encombrement et la consommation.

Les principales caractéristiques de ces nouvelles familles sont disponibles dans le catalogue joint, les datasheets complètes sont incluses dans la dernière version du CD Rom. Des informations de dernières minutes sont consultables sur le site de WSI [www.wsipsd.com](http://www.wsipsd.com)

Je me tiens à votre disposition pour toute information complémentaire et vous prie d'agréer, Madame, Monsieur, l'expression de mes salutations distinguées.

François MORIAMEZ  
Ingénieur d'Application



# FLASH PSD

**MCU  
Peripherals**

**MiROEL**

ISO 9002 AFAQ N° 1998/9648  
B.P. N°3 91941 LES ULIS CEDEX  
Tél. : 01 69 07 08 24 Fax. : 01 69 07 17 23  
[www.microel.fr](http://www.microel.fr) [microel@wanadoo.fr](mailto:microel@wanadoo.fr)

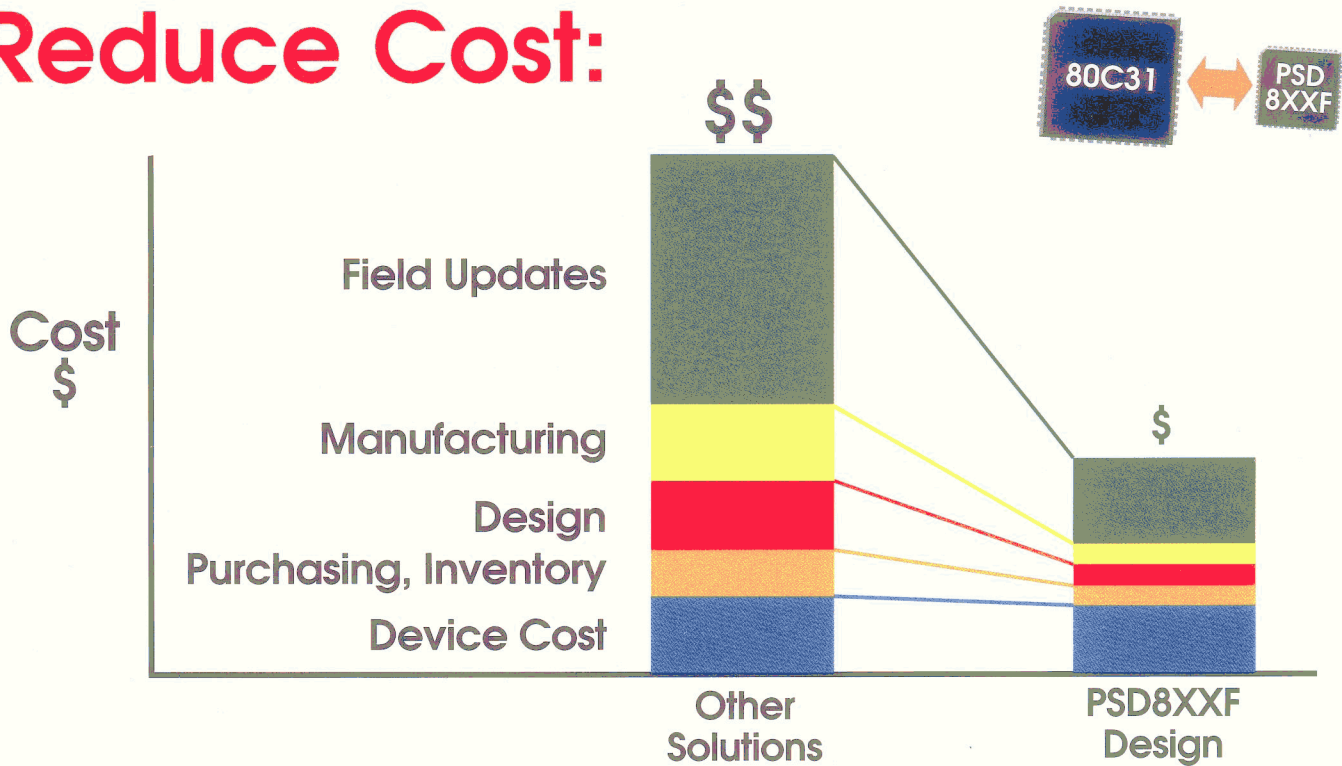


# PSD8XXF:

## a broad family of Flash MCU peripherals

Part		Logic					I/O	Memory				Other			
PSD	ZPSD	PLD inputs					Ports	Flash Program Store				J-TAG			
		Input Micro↔ Cells			Output Micro↔ Cells			2nd Flash Boot		EEPROM/EEPROM boot		ISP			
		PLD outputs						OTP EPROM boot		SRAM		Security			
		Page Reg										PMU			
PSD813FH		63	23	12	12	4	19	1024Kb			256Kb	4Kb	X	X	X
PSD813FN		63	23	12	12	4	19	1024Kb			256Kb	4Kb	X	X	X
PSD813F1	X	73	24	16	19	8	27	1024Kb		256Kb		16Kb	X	X	X
PSD813F2	X	73	24	16	19	8	27	1024Kb	256Kb			16Kb	X	X	X
PSD813F3	X	73	24	16	19	8	27	1024Kb				16Kb	X	X	X
PSD813F4	X	73	24	16	19	8	27	1024Kb	256Kb				X	X	X
PSD813F5	X	73	24	16	19	8	27	1024Kb					X	X	X

## Reduce Cost:



The **PSD8XXF** reduces cost in several areas. It can be used for first-time low-cost designs, mid-life product cost reduction, or operations expense reduction.

# Introducing

# FLASH PSD8XXF

## ZPLD Input Bus:

73 input signals for great system flexibility and control.

## MCU Interface:

Programmable interface to most 8-bit MCUs from Intel, Philips, Motorola, Zilog, Hitachi, and others.

## Your 8-bit MCU:

Use your low-cost, ROM-less 8-bit MCU with a Flash PSD8XXF and take advantage of the software you already have.

## Decode PLD:

73 inputs; selects Flash, EEPROM, and SRAM sectors, JTAG and peripheral selects; performs complex mapping which enables non-stop operation during the Flash memory reprogramming cycle.

## JTAG Serial Programming:

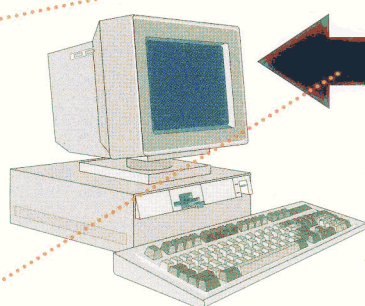
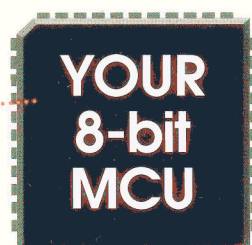
Enables in-system first-time programming; on-board command processor achieves 600% programming speed increase over parallel programming; MCU can also parallel program the PSD8XXF memory.

## ISP Loader:

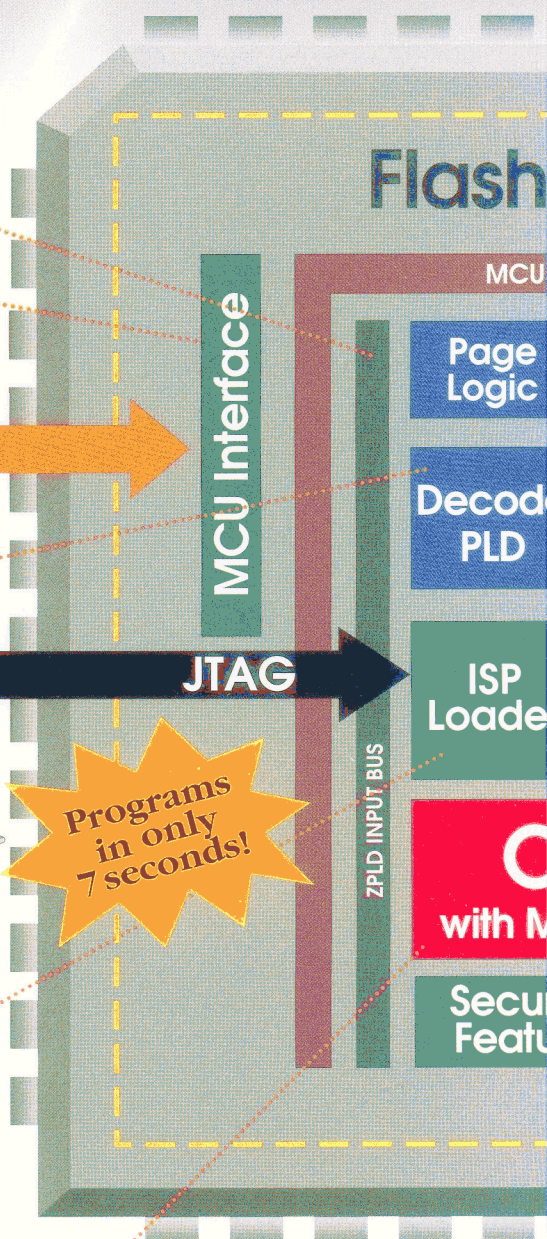
Self-contained JTAG command processor available at power-up; ultra-fast programming algorithm; state machine, data pipeline, etc. enables as fast as 7-second programming of the entire 100% ISP Flash PSD8XXF.

## CPLD:

3,000 gate equivalent; fast ISP programmable; direct R/W access between MCU and mapped Micro $\Rightarrow$ Cells; 73 inputs, 16 output and 24 input Micro $\Rightarrow$ Cells; use for dual processor interface, serial/parallel I/O, event counting, state machine, timing synchronization, etc.



PC or ATE



# XXF for MCUs

## Page Logic:

8-bit page register expands your MCU addressing range by 256 times. Great for extended mapping.

## Concurrent Flash:

5-volt only, 128K x 8; eight 16K-byte mappable sectors; sector erase and protection; fast Serial JTAG or parallel byte-by-byte programming.

## EEPROM (F1):

5-volt only, 32K x 8; four mappable 8K-byte sectors; use for variable data, constants, or boot code; byte programmable and erasable. This second array is also available as OTP EPROM (FH/FN) or Flash (F2).

## Scratch-pad SRAM:

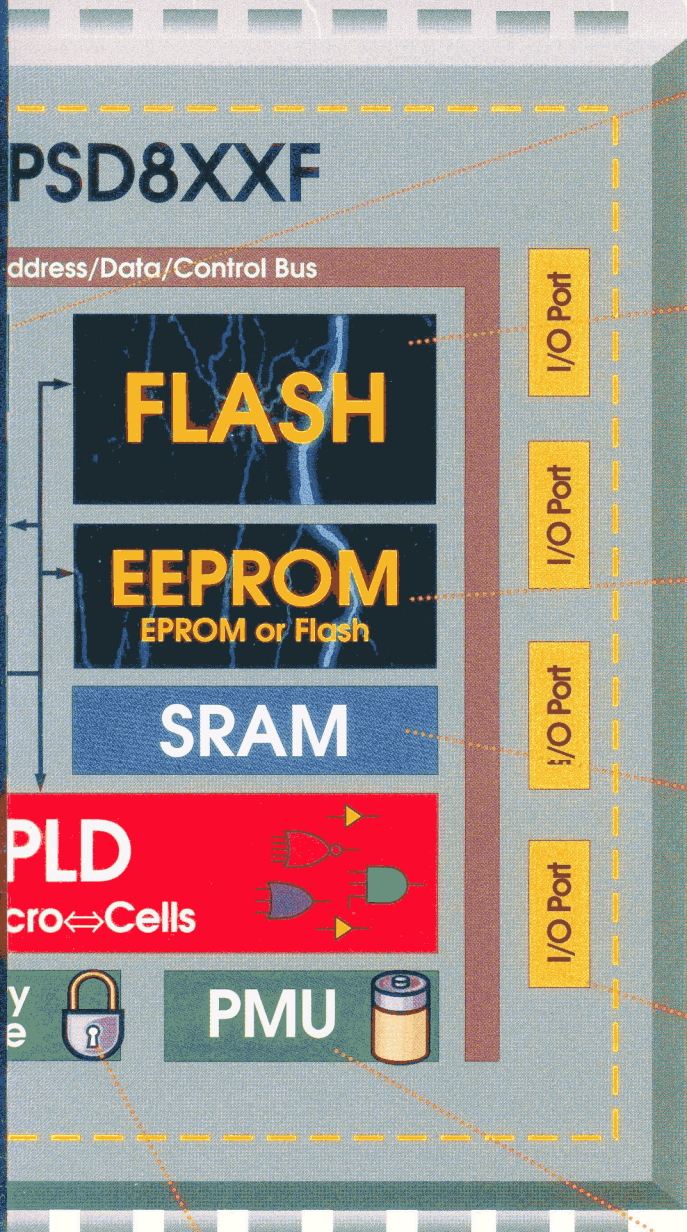
Battery backed, 2K x 8; automatic power switching when  $V_{CC} < V_{stdby}$ ; 0.5  $\mu$ A standby current.

## I/O Ports:

27 individually programmable I/O ports; replace lost I/O on your ROM-less MCU; unlimited possibilities when used with the CPLD.

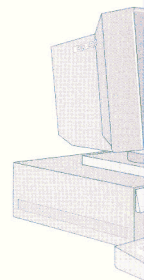
## Zero-Power Unit:

Optimizes AC and DC power consumption; Zero-Power features provide automatic power down, standby (25 $\mu$ A typical), or sleep modes. Typical active power is only 1 mA/MHz (bus frequency)... just 7% of the power required by a discrete solution!



## Design Security:

Locks up your program code and configuration information from your competition.



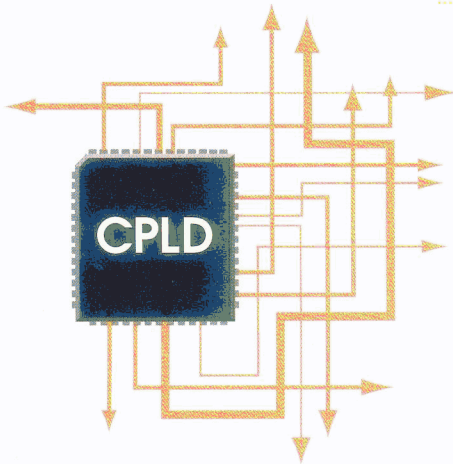
# Easy FLASH™

**FLASH**

**Second Memory Array  
EEPROM, Flash, EPROM**

## Get non-stop concurrent operation with Flash and a second memory array

While the Flash memory is busy with Erase or Write functions, your MCU can execute code from the second memory array (EEPROM, Flash, or EPROM). This prevents risky SRAM operation, long MCU wait states, and system hang-ups.



## PSD8XXF automatically performs complex mapping

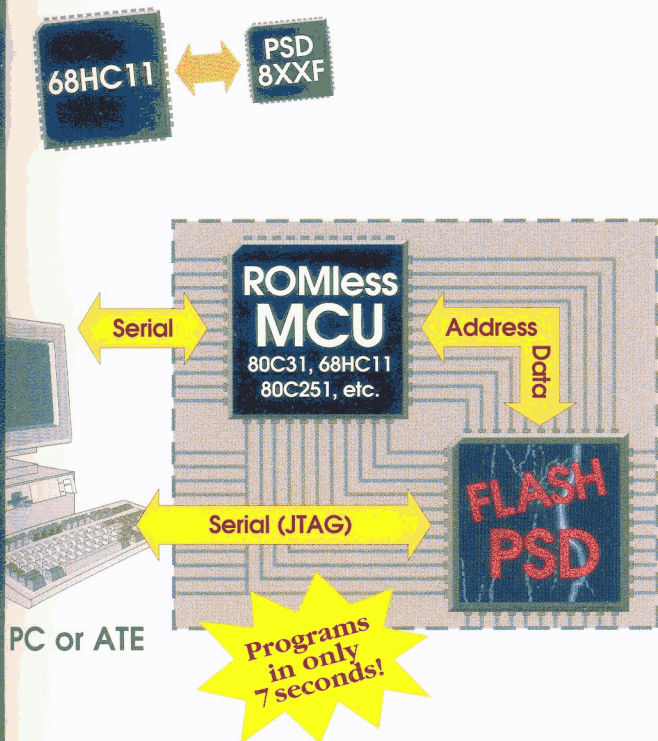
Flash program code must be remapped to data space when being updated. The MCU must operate from other program code during that time. Multiple product terms in the decode PLD, page logic, and run-time control registers permit dynamic remapping and non-stop MCU operation during Flash updating.

## 100% In-System Programmable... even the first time!

The PSD8XXF's on-board JTAG command interpreter is ready at power-up on a blank device. The Flash and EEPROM memory, decode PLD, CPLD, I/O, MCU interface, etc. are all serial JTAG programmable in as little as 7 seconds. No need for a ROMed MCU with boot-code or a separate device programmer. Get the WSI Flash Link serial programming accessory and program right from your PC.

## PSDsoft makes it easy

PSDsoft software tools make fast, easy work of configuring and programming PSD8XXF to try different logic designs and prototype iterations. These tools even create ANSI-C code for your MCU to access, control and program the PSD8XXF. **We make it easy.**



## For more information

✓ Email WSI: [info@wsiusa.com](mailto:info@wsiusa.com)

✓ Call WSI: 800-TEAM-WSI (832-6974)

✓ Call your local WSI Sales Rep or Distributor

✓ Return the postage-paid card

✓ Visit [www.wsipsd.com](http://www.wsipsd.com)

✓ Fax the card to WSI: (510) 657-8495



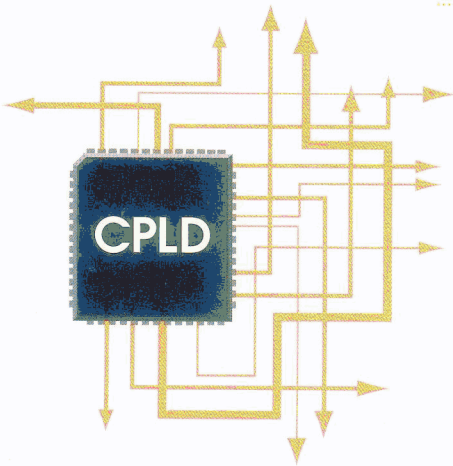
# Easy FLASH™

**FLASH**

**Second Memory Array  
EEPROM, Flash, EPROM**

## Get non-stop concurrent operation with Flash and a second memory array

While the Flash memory is busy with Erase or Write functions, your MCU can execute code from the second memory array (EEPROM, Flash, or EPROM). This prevents risky SRAM operation, long MCU wait states, and system hang-ups.



## PSD8XXF automatically performs complex mapping

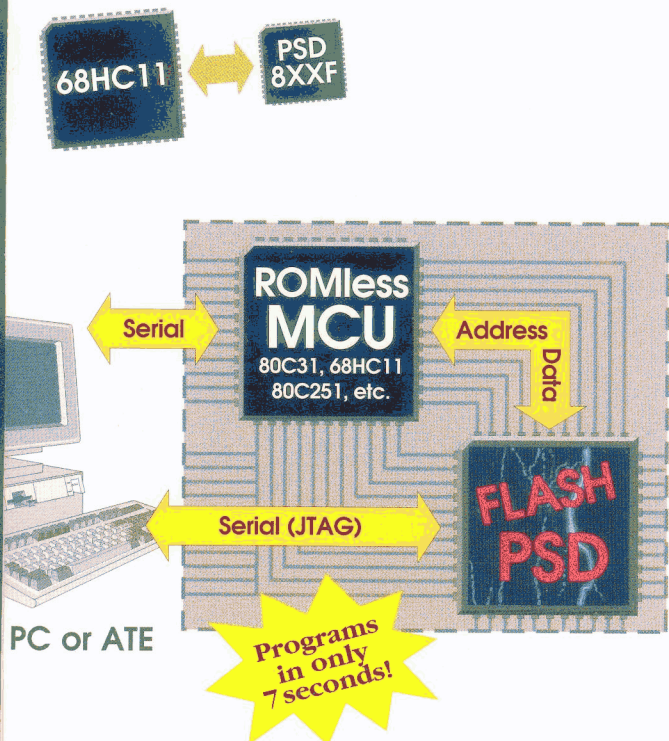
Flash program code must be remapped to data space when being updated. The MCU must operate from other program code during that time. Multiple product terms in the decode PLD, page logic, and run-time control registers permit dynamic remapping and non-stop MCU operation during Flash updating.

## 100% In-System Programmable... even the first time!

The PSD8XXF's on-board JTAG command interpreter is ready at power-up on a blank device. The Flash and EEPROM memory, decode PLD, CPLD, I/O, MCU interface, etc. are all serial JTAG programmable in as little as 7 seconds. No need for a ROMed MCU with boot-code or a separate device programmer. Get the WSI Flash Link serial programming accessory and program right from your PC.

## PSDsoft makes it easy

PSDsoft software tools make fast, easy work of configuring and programming PSD8XXF to try different logic designs and prototype iterations. These tools even create ANSI-C code for your MCU to access, control and reprogram the PSD8XXF. **We make it easy.**

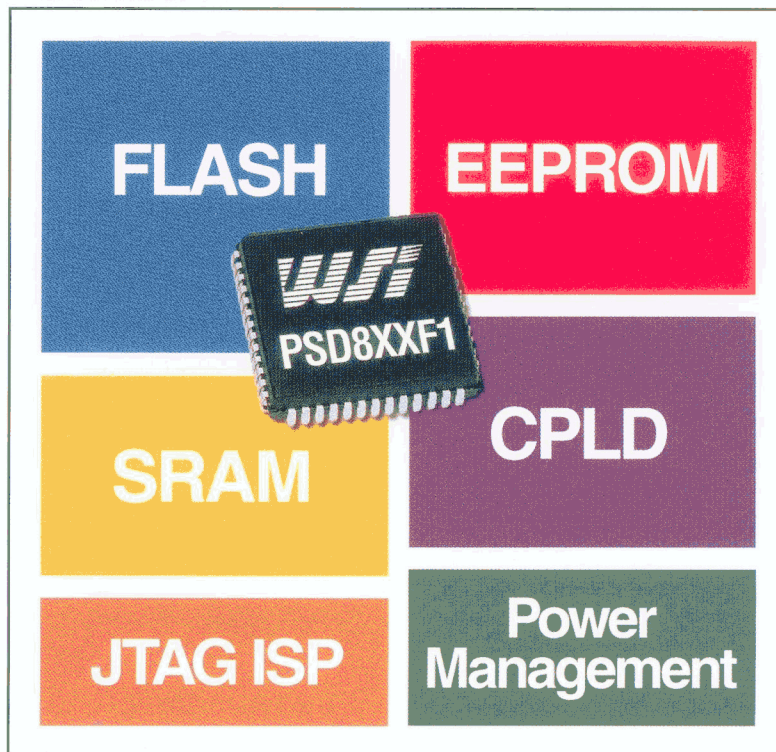


**For more information**

- Return the postage-paid card
- Visit [www.wspisd.com](http://www.wspisd.com)
- Fax the card to WSI: (510) 657-8495
- Email WSI: [info@wsiusa.com](mailto:info@wsiusa.com)
- Call WSI: 800-TEAM-WSI (832-6974)
- Call your local WSI Sales Rep. or Distributor



# Well, now you can!

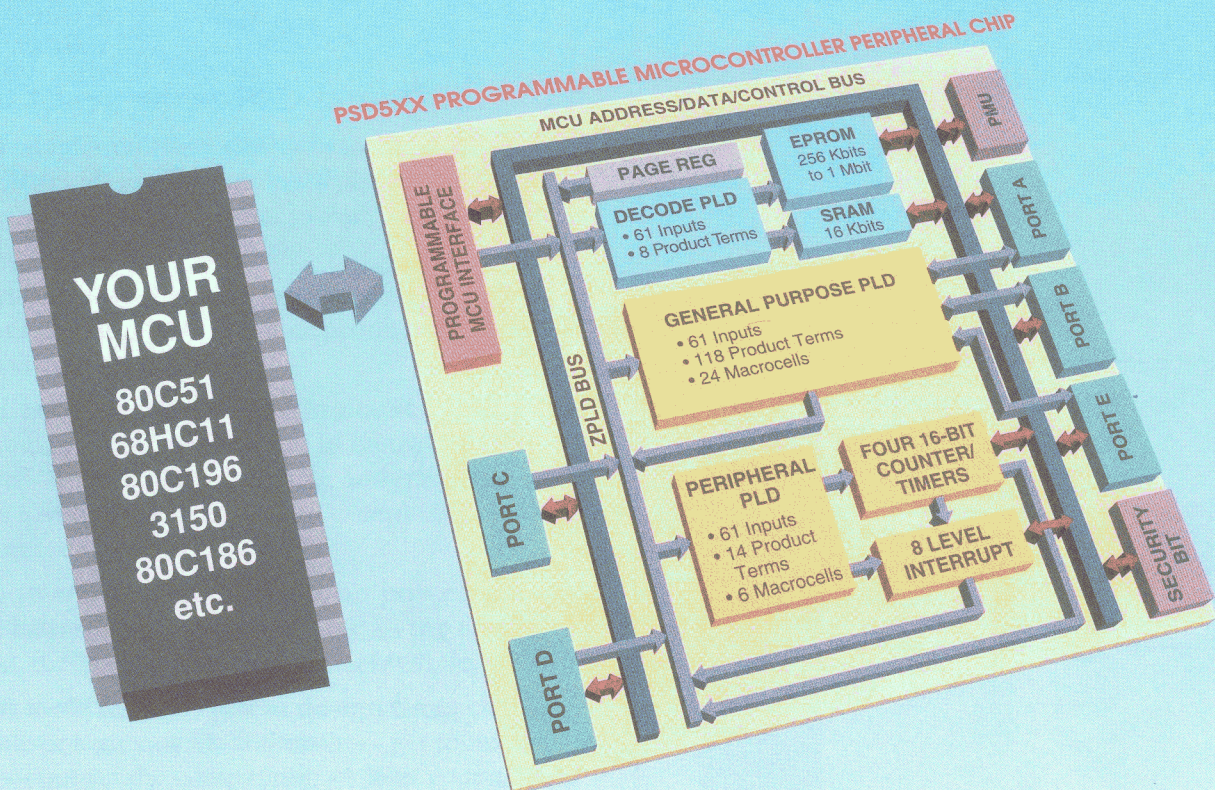


Introducing

# *Easy* FLASH™

# PSD™

## Programmable Peripherals for Microcontrollers



# Save Power, Product size, Design time and Money!

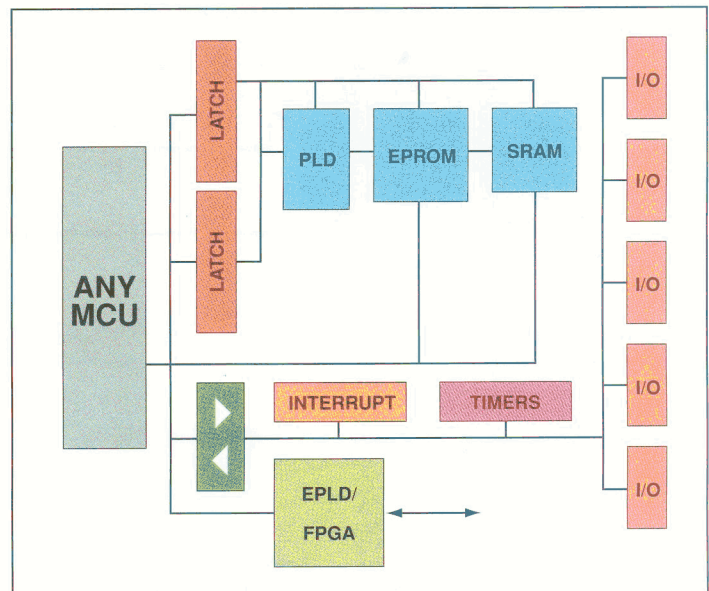
**W**hen your design requirements exceed the capability of your MCU, add a PSD micro-controller peripheral IC and get many more resources, simplify the design process, and cost reduce your system.

Different PSDs provide combinations of functions needed for efficient embedded control: programmable logic, EPROM, SRAM, programmable I/O ports, counters, timers, interrupt controller, power management, design security, and much more.

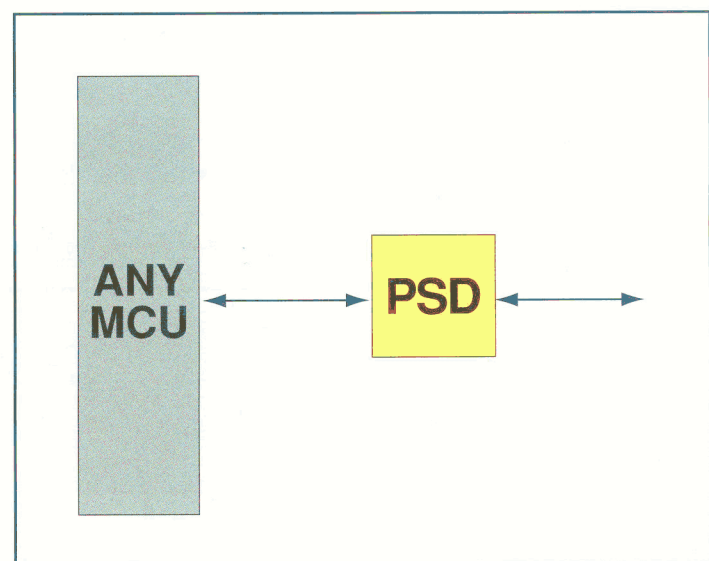
## Look at these great PSD benefits!

- **PSD programmability delivers great design flexibility:** PSD reconfigurability gives you changeable design options — adapts to your needs — solves YOUR design problems quickly.
- **PSD system integration reduces product size:** customers prefer smaller, lighter products that do more and cost less.
- **Minimize your manufacturing costs:** one programmable PSD can be used in many different products — simplifies purchasing and handling, reduces inventory and P.C. board complexity. PSDs also increase system reliability.
- **Zero-power PSDs take almost no power:** use smaller batteries — extend their life — the longer they last, the happier your customers will be.
- **PSDsoft tools reduce system design time:** change your code or logic quickly and easily — get to market sooner — outrun the competition — **lead** your market.
- **Hide your design from prying eyes:** the PSD security bit locks in your firmware — competition can't copy your design.
- **Low EMI:** very low power and fewer board traces mean low, low EMI from your design — fly right through emission testing.

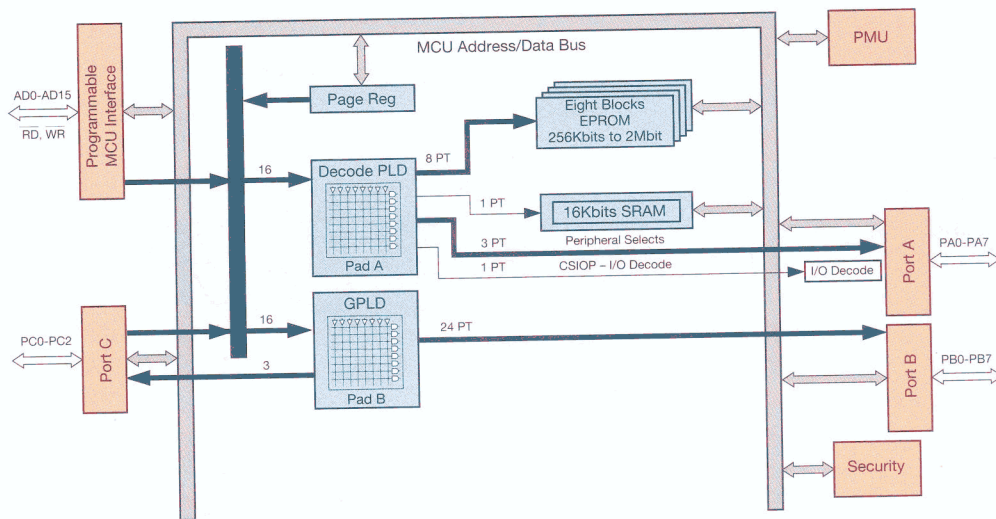
## Before



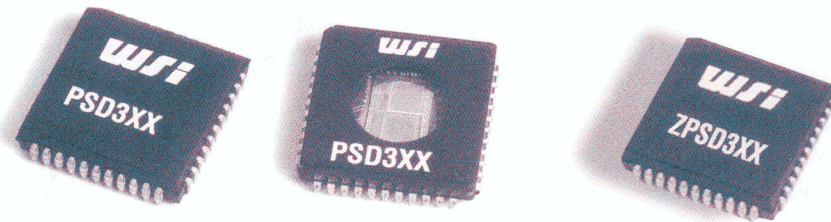
## After



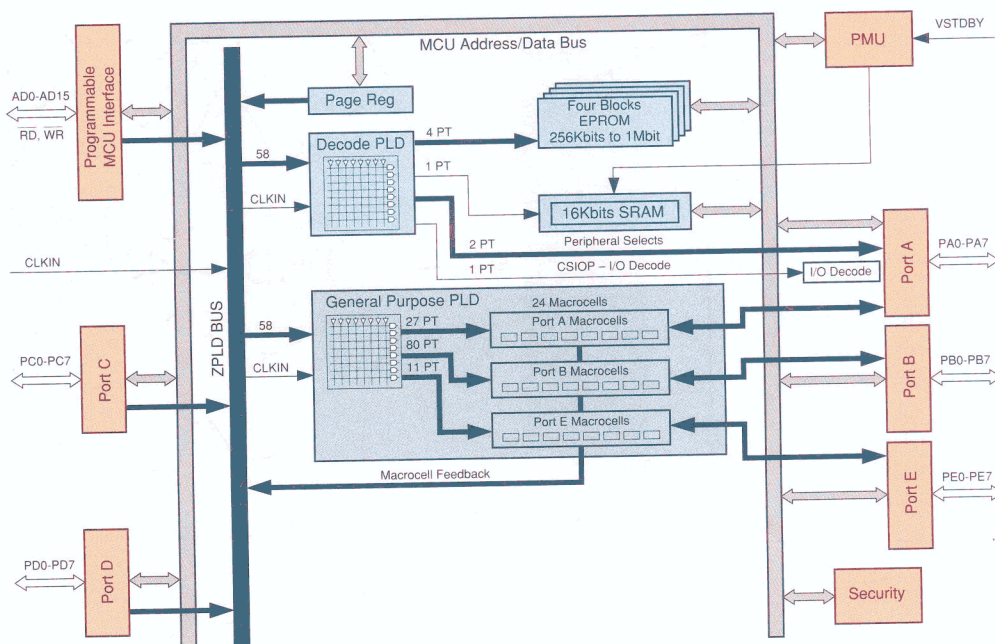
# The Three PSD Families



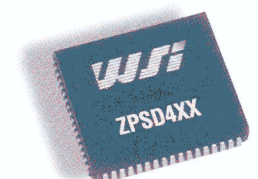
The **PSD3XX** and **ZPSD3XX** Families include 256Kb to 2Mb of EPROM, 16 Kb of SRAM, programmable decode logic for the internal memory and external chip selects, a 4-bit page register for bank switching, 19 programmable I/O port pins, a programmable microcontroller interface, and a programmable security bit that protects your design from the competition. SRAMless versions are available for designs requiring large external SRAMS.



The Zero-Power Technology includes **Integrated Power Management** which controls the world's first zero power PLD, EPROM and SRAM memory.

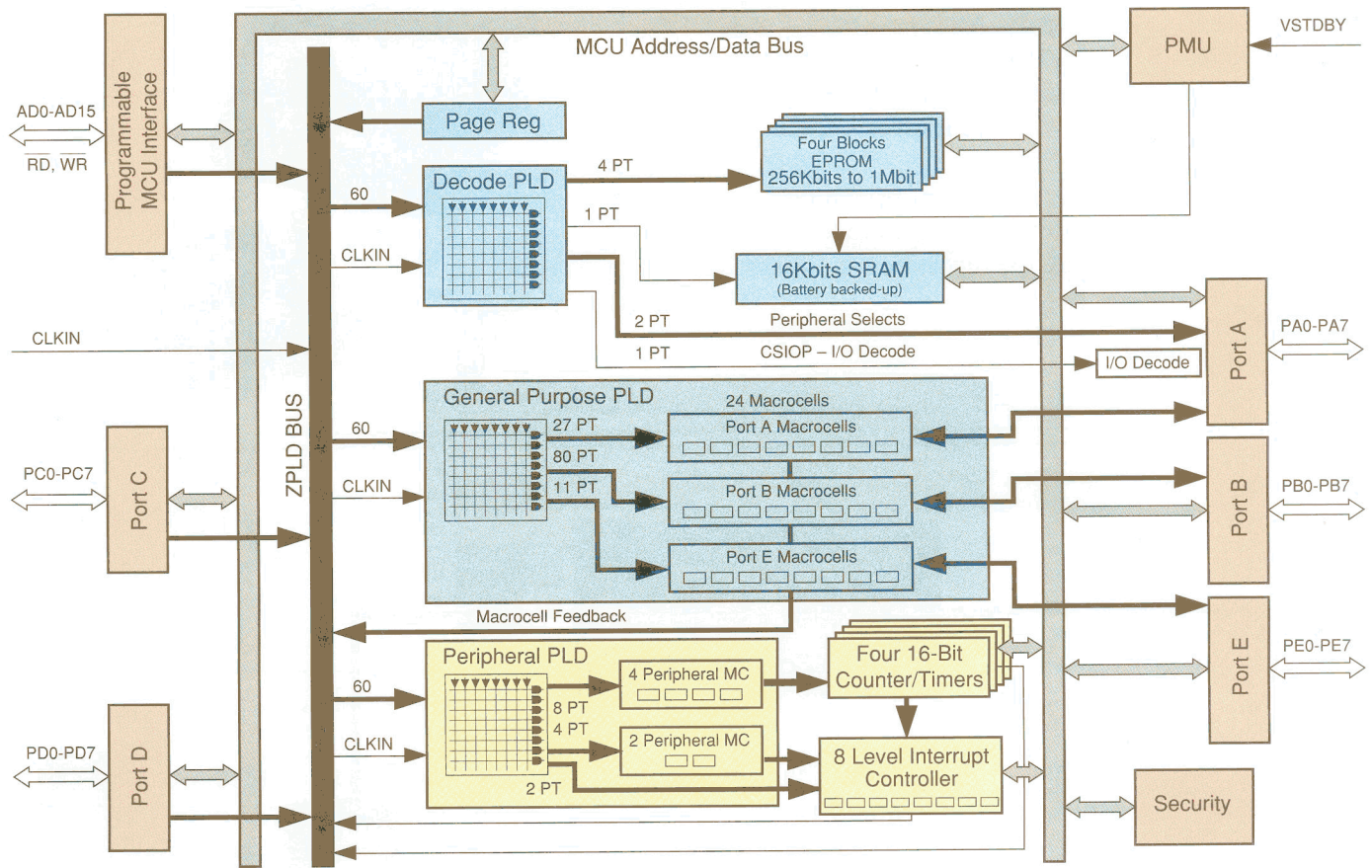
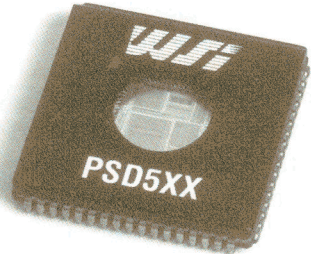
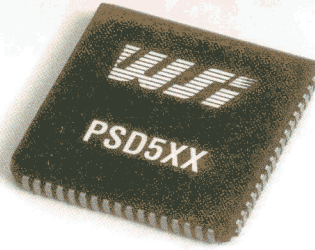


The **PSD4XX** and **ZPSD4XX** Families add several features including a general-purpose PLD (GPLD) with 59 inputs and 118 product terms in 24 flexible macrocells. A power management unit controls the zero power PLDs, the battery backed SRAM and the automatic power down unit. Additional I/O ports give a total of 40 programmable I/O pins.



## PSD5XX and ZPSD5XX Family

devices add several features to the PSD4XX architecture. Four 16-bit counter/timers including a watchdog timer and pulse width modulator are provided along with an eight-level priority interrupt controller. The Peripheral PLD (PPLD) controls the counter/timers and interrupt controller. Total zero-power PLD resources include 61 inputs, 140 product terms and 30 registered macrocells.



The PLD arrays enable you to control the internal functional blocks and efficiently implement a variety of state machines, logic functions, and address decoders. User-definable features include Data Bus width (8- or 16-bits), Address Bus width, I/O pin functions, MCU interface and more. The General Purpose PLD (GPLD) and the Peripheral PLD (PPLD) operate independently from your microcontroller.

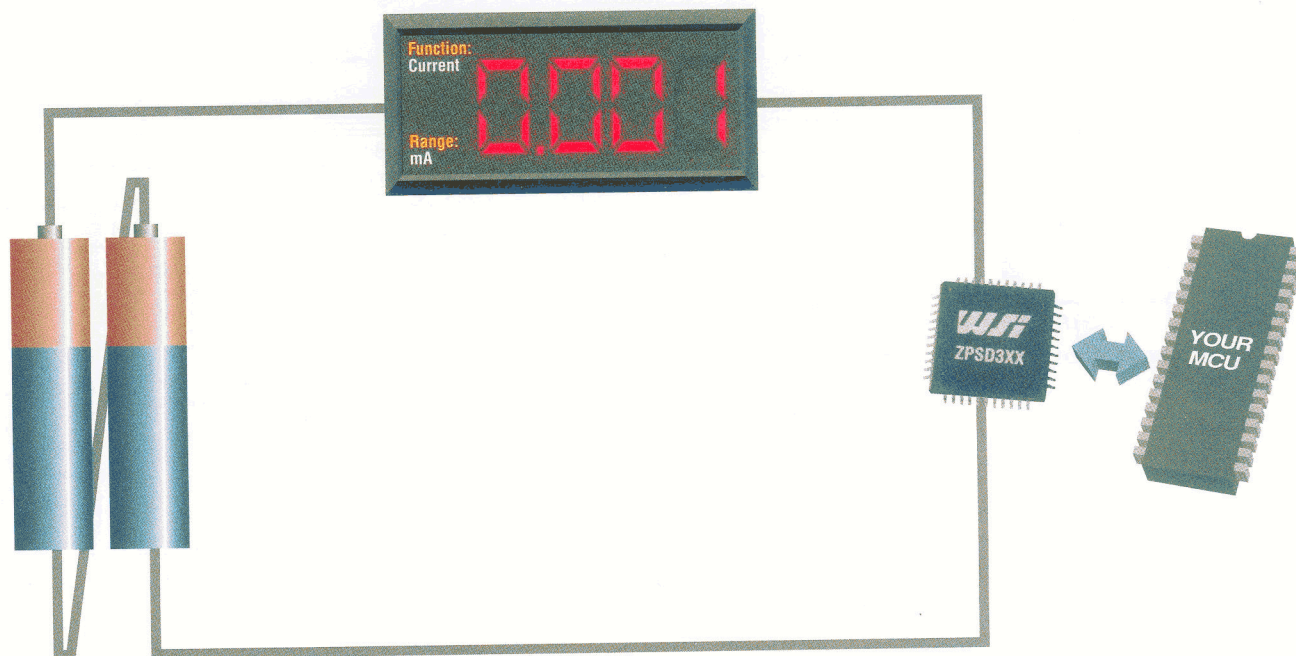
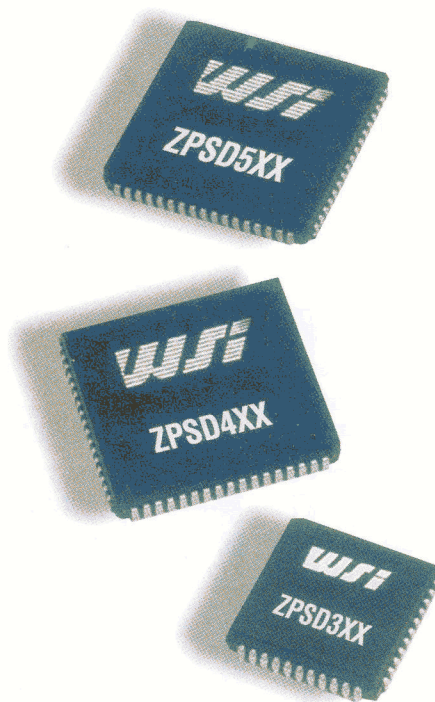
# ZPSD: Zero-Power Programmable MCU Peripherals

The PSD3XX / 4XX / and 5XX architectures are also available in Zero-Power ZPSD versions. They are the ultimate in low power solutions for use in battery powered portable products.

In standby, 2.7 volt or 5.0 volt ZPSDs typically draw only 1 or 10 microamps respectively. During operation, the ZPSD's EPROM requires only 0.4 or 0.8 mA per bus MHz respectively.

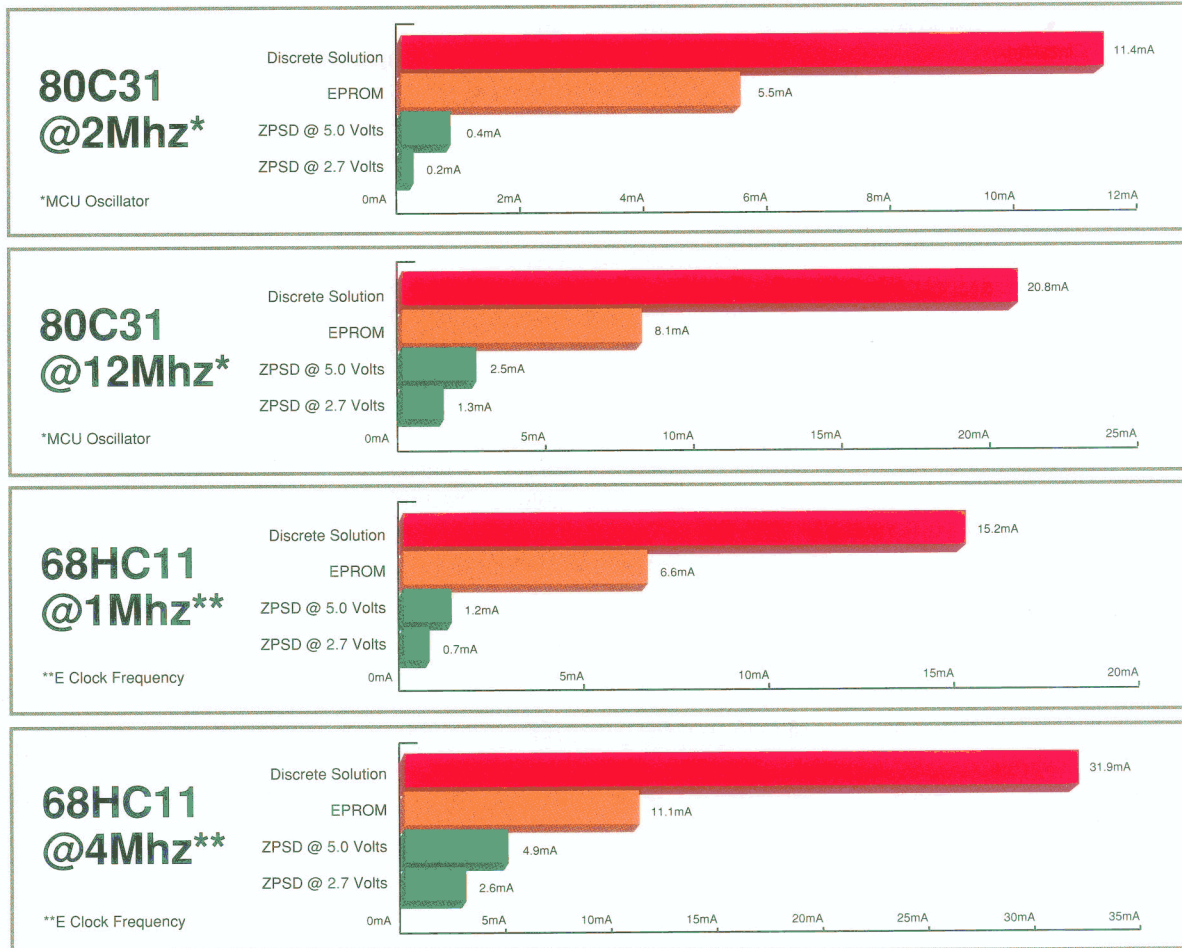
For example, a 5.0-volt ZPSD3XX consumes only 0.8 mA operating current when used with a 4-MHz 8031. This is an 8 to 16-fold improvement over a discrete solution!

Best of all, this incredible low power operation is automatic, will not slow down your system, and requires no special hardware or software.



# ZPSD Integrated Power Management Technology

Use the ZPSD with virtually any 8 or 16 bit MCU and compare active power requirements with other low power peripheral solution!



Note: Assumes EPROM is accessed 80% of the time, SRAM @ 15%, and I/O @ 5%; Published data sheet values.

Every aspect of the ZPSD has been designed to minimize power consumption. All internal sections of the device remain automatically powered down unless actually being accessed... all without any performance degradation. Check out these low power design innovations...

## Address Transition Detection (ATD)

When the ATD circuitry detects an address change, it proceeds to power up the internal sections. It generates sequential clocking for the address decoder, memory arrays, sense amplifiers, and the output data latch. Then the ZPSD automatically reverts back to standby mode.

## Differential Pass Transistor Logic

Pass transistors and differential logic reduce the internal signal voltage swing to only 2.5 volts. Pass transistors contain no P-channels, have 70% less capacitance and pass the signal directly to the output without any connection to the power supply. The payoff is low, low power consumption.

## Dual Reference Sense Amplifier (DRSA)

Low power DRSA technology uses a differential sensing scheme that detects the direction of signal voltage change rather than absolute value. A signal as low as 63 mv can be sensed by the DRSA. Small signal operation equals very low power.

# The PSD Families

## PSD3XX

PSD3XX devices include 256K to 2048 Kbits of EPROM; 16 Kbits of SRAM; programmable decode logic for the internal memory and external chip selects; a 4-bit page register; 19 programmable I/O port pins; and a programmable microcontroller interface.

Part Number	Bus Width	DPLD		I/O Port Pins	EPROM Kbit	SRAM Kbit
		Inputs	Product Terms			
PSD301R	x8 / x16	12	40	19	256	
PSD311R	x8	12	40	19	256	
PSD302R	x8 / x16	16	40	19	512	
PSD312R	x8	16	40	19	512	
PSD303R	x8 / x16	16	40	19	1024	
PSD313R	x8	16	40	19	1024	
PSD304R	x8 / x16	16	40	19	2048	
PSD314R	x8	16	40	19	2048	
PSD301	x8 / x16	12	40	19	256	16
PSD311	x8	12	40	19	256	16
PSD302	x8 / x16	16	40	19	512	16
PSD312	x8	16	40	19	512	16
PSD303	x8 / x16	16	40	19	1024	16
PSD313	x8	16	40	19	1024	16

(R-suffix = SRAMless)

## PSD4XX

PSD4XX devices add a general-purpose, high-density PLD with 59 inputs and 118 product terms in 24 flexible macrocells. A power management unit controls the zero power PLDs, battery back-up SRAM and the automatic power-down unit. Additional I/O ports give a total of 40 programmable I/O pins.

Part Number	Bus Width	DPLD + GPLD			I/O Port Pins	PMU	EPROM Kbit	SRAM Kbit
		Inputs	Product Terms	Registered Macrocells				
PSD401A1	x8 / x16	37	113	8	40	Yes	256	16
PSD411A1	x8	37	113	8	40	Yes	256	16
PSD402A1	x8 / x16	37	113	8	40	Yes	512	16
PSD412A1	x8	37	113	8	40	Yes	512	16
PSD403A1	x8 / x16	37	113	8	40	Yes	1024	16
PSD413A1*	x8	37	113	8	40	Yes	1024	16
PSD401A2	x8 / x16	59	126	24	40	Yes	256	16
PSD411A2	x8	59	126	24	40	Yes	256	16
PSD402A2	x8 / x16	59	126	24	40	Yes	512	16
PSD412A2	x8	59	126	24	40	Yes	512	16
PSD403A2	x8 / x16	59	126	24	40	Yes	1024	16
PSD413A2*	x8	59	126	24	40	Yes	1024	16

\* Flash memory versions available.

## PSD5XX

PSD5XX devices add four 16-bit counter/timers including a watchdog timer; an eight-level priority interrupt controller; and a peripheral PLD to control the counter/timers and interrupt controller.

Part Number	Bus Width	DPLD + GPLD + PPLD			I/O Port Pins	PMU	EPROM Kbit	SRAM Kbit	Counter/Timer	Interrupt Controller
		Inputs	Product Terms	Registered Macrocells						
PSD501B1	x8 / x16	61	140	30	40	Yes	256	16	4-16 bit	8-Level
PSD511B1	x8	61	140	30	40	Yes	256	16	4-16 bit	8-Level
PSD502B1	x8 / x16	61	140	30	40	Yes	512	16	4-16 bit	8-Level
PSD512B1	x8	61	140	30	40	Yes	512	16	4-16 bit	8-Level
PSD503B1	x8 / x16	61	140	30	40	Yes	1024	16	4-16 bit	8-Level
PSD513B1	x8	61	140	30	40	Yes	1024	16	4-16 bit	8-Level

The majority of the above listed devices are also available as 5-volt zero-power devices (Example: ZPSD311) as well as 2.7 volt zero-power devices (Example: ZPSD311V)

PSDs and ZPSDs are available in three selections: erasable/reprogrammable windowed packages, plastic OTP versions, and lowest-cost mask programmable versions.

# PSD Products are Field-Proven

PSD programmable microcontroller peripherals are the ideal solution whenever functionality, low power, small product size, low manufacturing cost, and rapid

system development are important. PSDs are hard at work worldwide in hundreds of Fortune 1000 companies in a wide variety of applications.

## A Few Actual PSD Design Applications



Cellular Phone

ISDN Cards

Network Hubs

2-way Radios

Video CD Players

Set Top Boxes

Gas Pump Control

Telephone Echo Canceling

Blood Analyzer

Facsimile Machine

Internal FAX/Modem

A.C. Motor Control

Electronic Scales

Office Copier

CATV Repeater

LAN Board

Remote Utility Meters

Audio Sound Board

Fleet Tracking

Home Security System

Barcode Reader

Intelligent Pager

Cardiac Monitor

Retinal Scanner

Cable T.V. Decoder

Police Radar

Sonobuoy

Coin Validation

Private Branch Exchange (PBX)

Oven Control

Secure Communications

Fork Lift Control

Helicopter Guidance

Touch Screen Controller

USB Applications

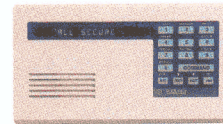
Pressure Measurement

Vehicle Brake Control

Elevator Door Control

Wheel Balancer

Ultrasound Imaging



RFID

Ocular Laser Control

Gas Detection

Radio Tuning Controller

Infrared Missile Guidance

Process Control

Flight Simulator

Telephone Line Cards

Liquid Chromatograph

Smart Welder

Glue Applicator

Voice Recognition

Traffic Controller

Wire Bonder

Karaoke Machine

Remote Data Logger

I.V. Fluid Control

Range Finder

Programmable Power Supply

Rubidium Time Reference



# PSD - All The Microcontroller Perip

## Programmable MCU Interface

Can be configured to interface with most any 8- or 16-bit MCU without any additional glue logic. Includes address latches for multiplexed MCUs.

## Independent PLD BUS

Allows logic functions to be independent of the MCU operations. This enables the designer to off-load many MCU operations into hardware embedded in the PSD device.

## Independent MCU Address/Data Bus

Handles address and data between the MCU, EPROM, SRAM, the PLD bus and external I/O.

## Page Register

Expands the MCU's addressable memory by 16 times.

## Decode PLD (DPLD)

Decodes up to 61 inputs, including the address from the MCU, to select one of the four blocks of internal EPROM, internal SRAM, and I/O ports.

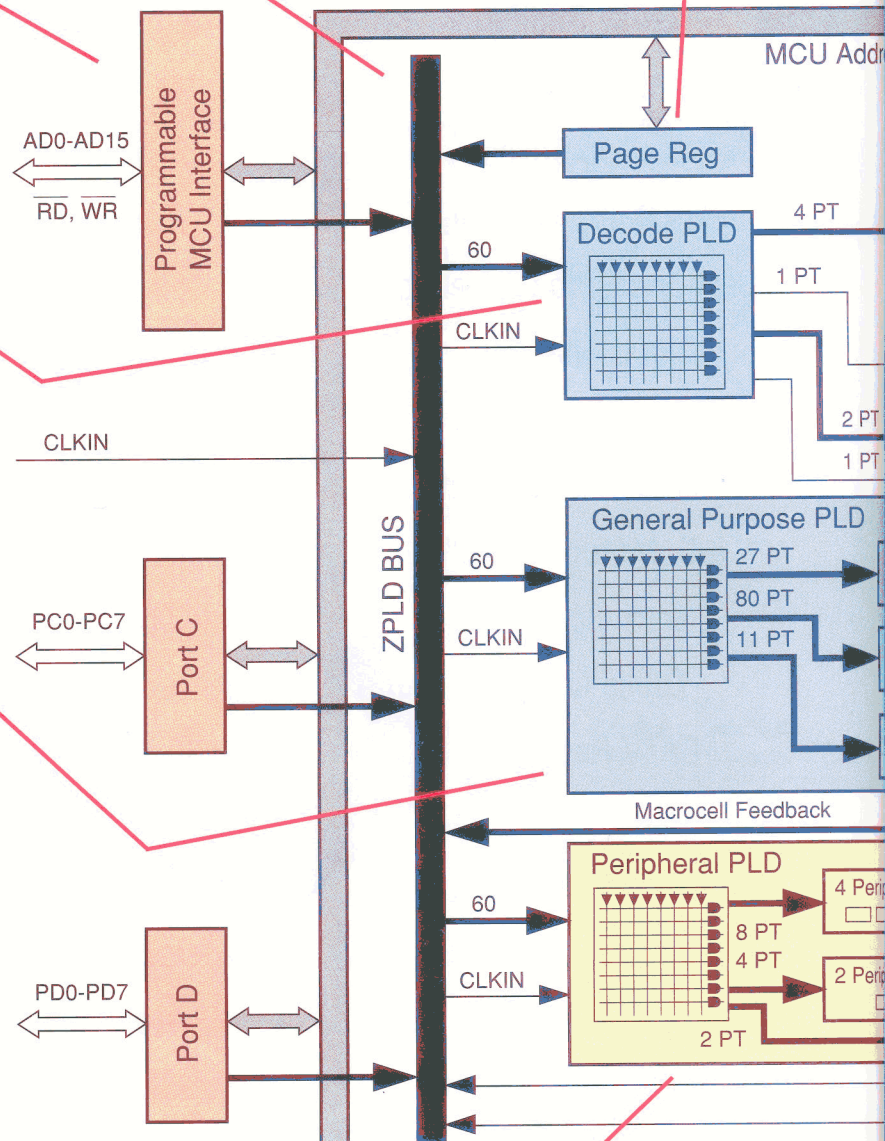
## General Purpose PLD (GPLD)

Up to 61 inputs and 118 product terms in 24 macrocells, and 24 outputs. Each macrocell is capable of registered or combinatorial outputs, as well as buried macrocell feedback. Can be used to implement a complex state machine; to select external devices; or to build serial ports, keypad controllers, or other combinatorial or sequential logic functions.

## Peripheral PLD (PPLD)

61 inputs and 14 product terms in 6 macrocells. Can be programmed to define conditions and events that control timing, counting and interrupt operations. These functions operate in parallel with MCU operations, so the MCU is free to perform other tasks.

## Block Diagram of



# Peripherals You Need In A Single Chip

## Up to 2 Mbit EPROM

Configurable by-8 or by-16. Fastest access time, including decode time and MCU interface delay, is 70 ns.

## Power Management Unit

Controls power consumption on the Z PSD device, including zero power PLDs (ZPLDs), EPROM power down modes, an automatic power-down counter, battery back-up for SRAM and C-Miser for memory.

## 16 Kbit SRAM

Augments the MCU's on-board register file. Has a dedicated Vstby supply pin for automatic battery back-up when Vcc drops 0.7 volts below Vstby.

## Five Programmable 8-bit I/O Ports

40 individually configurable I/O pins can be programmed as MCU I/O, PLD I/O, latched address outputs, open drain outputs, or I/O for several special functions. Each of the 40 I/O pins is individually configurable on-the-fly.

## Programmable Counter/Timers

Can simultaneously perform event-counting, time capture, watchdog, wave form, and pulse width modulation functions. Functions can be controlled by an input pin, a dedicated macrocell or by software. Each counter/timer is connected to the interrupt controller, the I/O ports, and the MCU bus. Since these firmware functions can be implemented in programmable hardware, software development and debug time can be reduced and the MCU can be freed to perform other tasks.

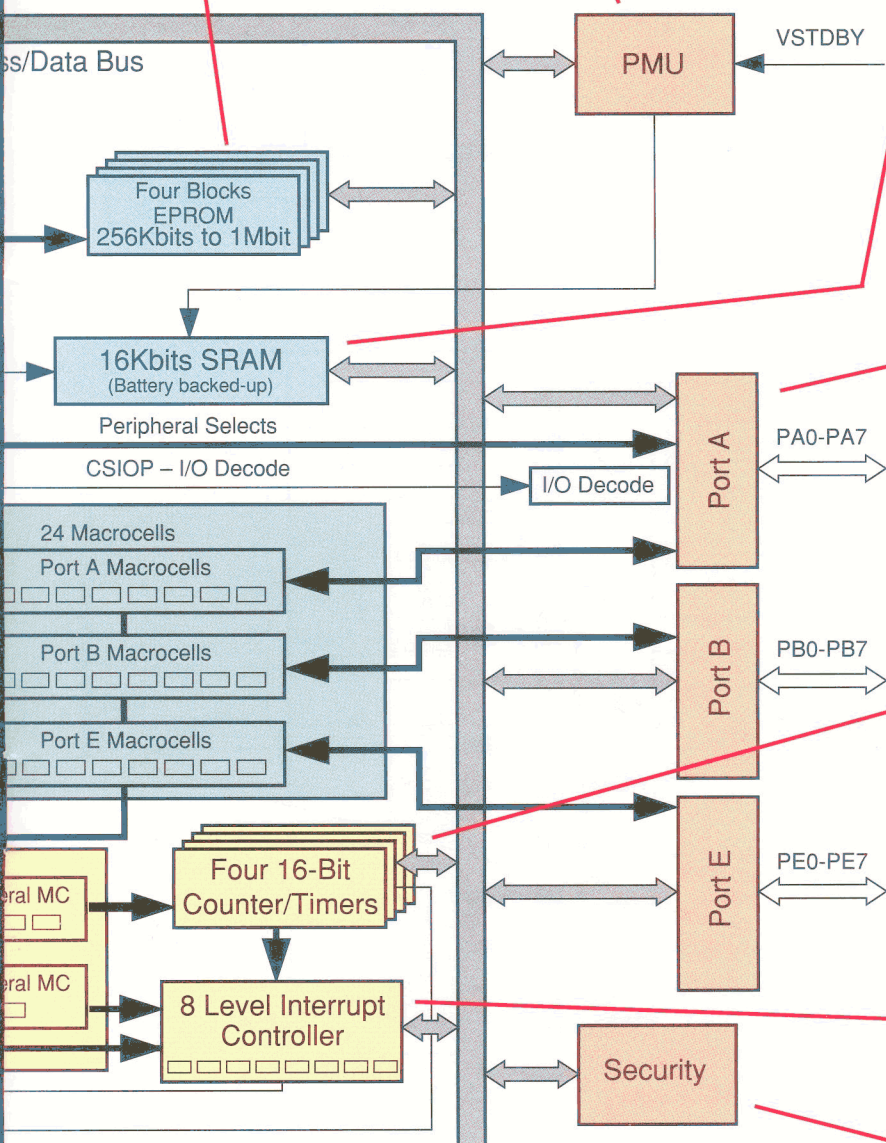
## Programmable Interrupt Controller

Can edge or level sense, mask, identify and decode priorities for eight internal interrupts generated by four dedicated macrocell outputs from the peripheral PLD or from four terminal-count outputs from the counter/timer unit.

## Security Bit

Locks device configuration to prevent unauthorized duplication.

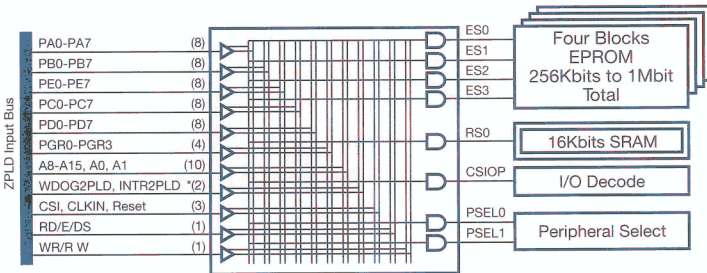
## the PSD5XXB1



# The PSD Families - Flexibility

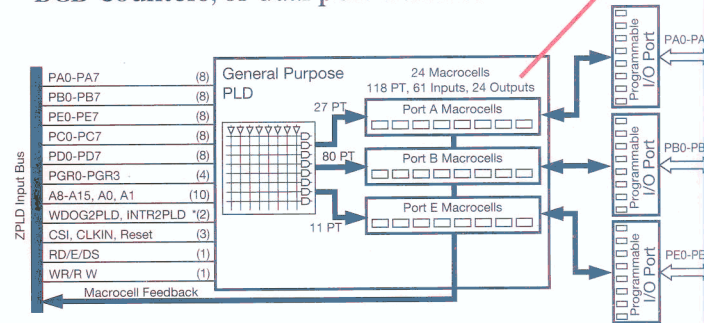
## Decode PLD

The decode PLD allows the easy mapping of internal functions to the MCU address space.

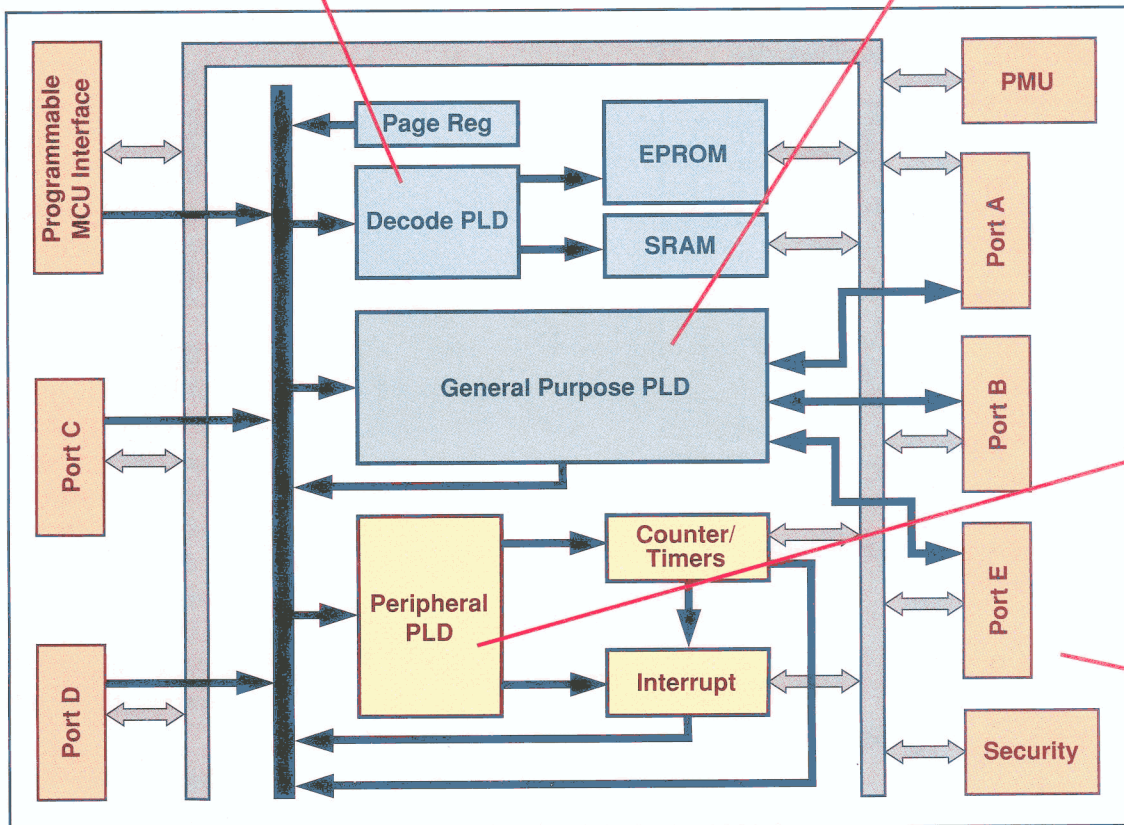


## GPLD

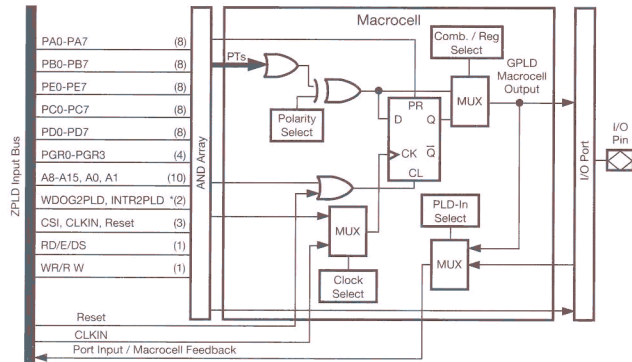
Has up to 61 inputs and 118 product terms in 24 macrocells, and 24 outputs. It can be used to develop such applications as standard or proprietary bus interfaces, serial channels, state machines, external address decoders, Manchester encoders, shaft encoders, stepper motor controllers, dynamic memory controllers, BCD counters, or dual port arbiters.



## PSD5XX

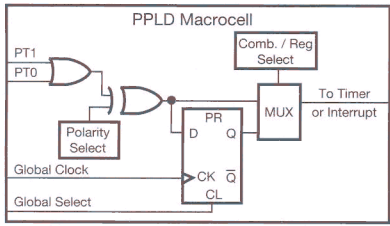


# Through Programmability



## GPLD Macrocell

Can be used as a buried feedback or for product term expansion. The combined features of the GPLD macrocell and I/O ports provide up to six flip-flops that can operate independently.



## Peripheral PLD (PSD5XX)

Has 61 inputs and 14 product terms with 6 macrocells. Four macrocells generate user-defined triggers for time or event capture functions in the counter/timer unit.

## Peripheral PLD Macrocell

Four macrocells for the four 16-Bit counter/timers and two macrocells for interrupt number 6 and 7.

## Counter/Timer (PSD5XX)

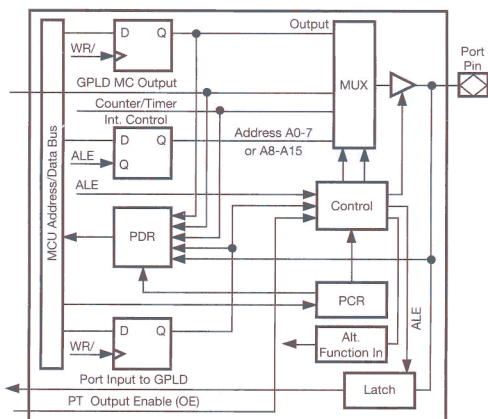
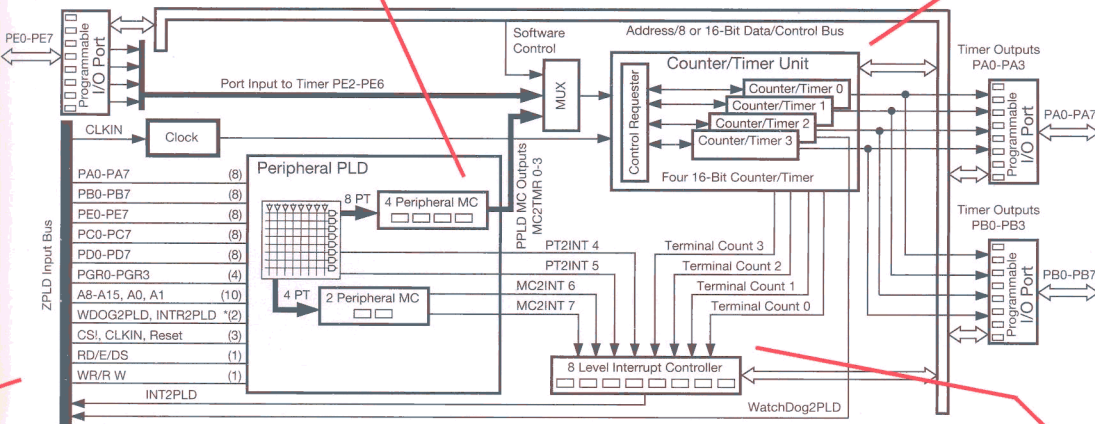
The PSD5XX has four 16-bit programmable and scalable counter/timers with a clock input of up to 30 MHz that can be divided from 4 to 280. Clock speed can be increased up to 40 MHz for each counter/timer when used with the GPLD.

The five modes of operation are:

- waveform
- pulse
- event counter
- time capture
- watchdog

## Interrupt Controller (PSD5XX)

The individually maskable, eight-level, automatic priority interrupt controller features four inputs from the peripheral PLD, four inputs from the peripheral counter/timer terminal counts and global interrupt output to Port E.

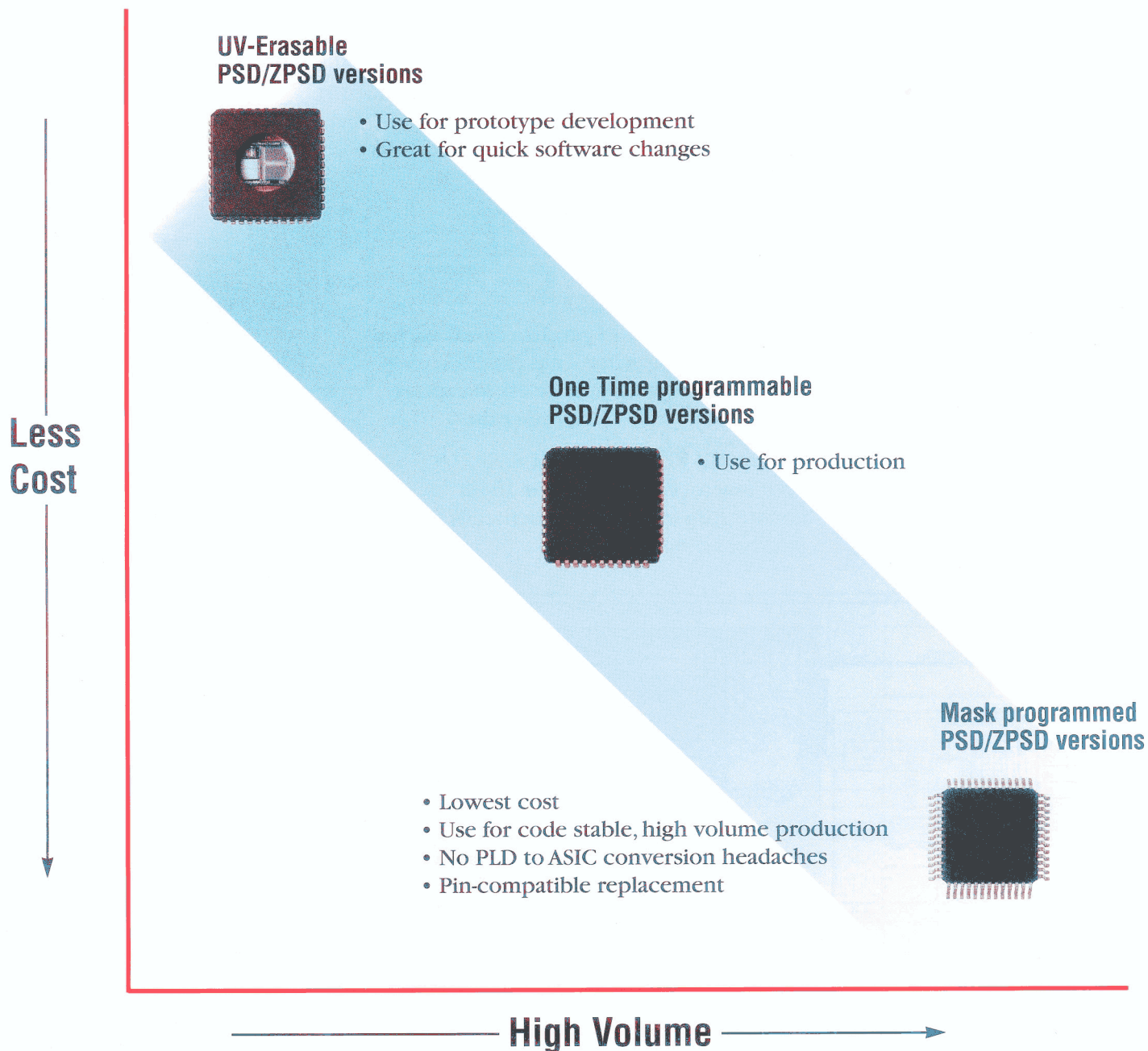


## PSD Port I/O Pins

Have a variety of options that are configurable on-the-fly as:

- MCU I/O ports
- PLD I/O ports
- Address I/O ports
- Counter/timer I/O ports
- Interrupt controller I/O ports

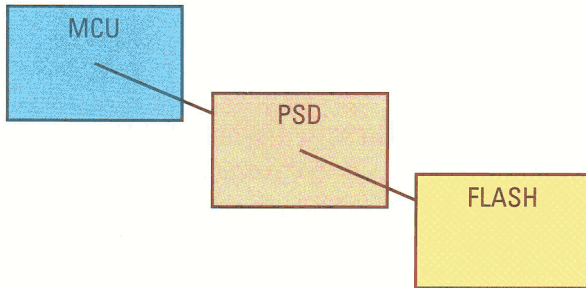
# Take the PSD path to low cost manufacturing



## Seamless, pin-compatible PSD cost reduction path from uv-erasable to masked versions

- Common architecture
- Common development tools
- Pinout and code compatible from prototype to high volume production
- Zero-Power, 2.7 or 5-volt mask programmed ZPSD peripherals for MCUs
  - Lowest Power!
  - Lowest Cost!

# Typical PSD Applications

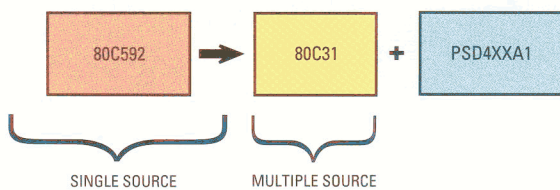
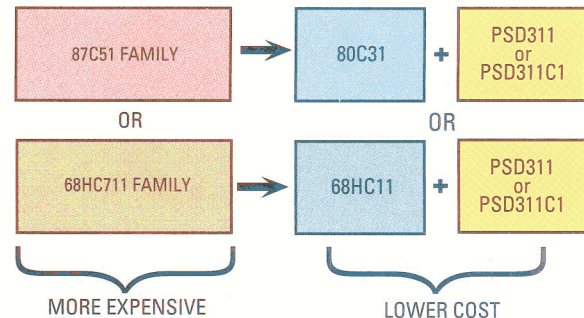


## Simplify Flash Memory Overhead

Place your boot code and programming algorithms inside the PSD non-volatile EPROM. This simplifies Flash memory reprogramming and secures the boot code. Eeprom based boot code also increases system reliability since a power glitch during Flash reprogramming may result in uncertain data. By reprogramming the Flash memory from the PSD after such an event, system reliability is assured. This technique does not require the MCU to move the programming algorithms to SRAM for execution.

## Eliminate Costly OTP MCUs

When used with an inexpensive generic 8-bit MCU, a low cost PSD device enables you to significantly reduce your system cost. In addition, you gain much more system control capability.

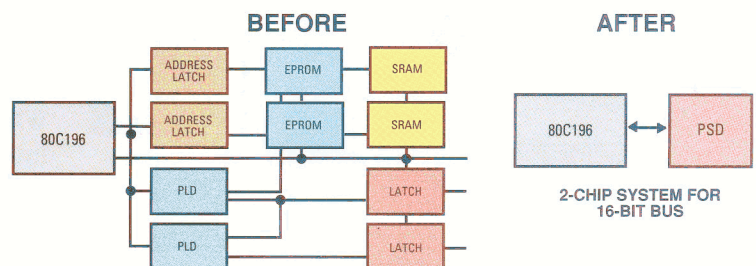


## Eliminate Sole-Sourced MCUs

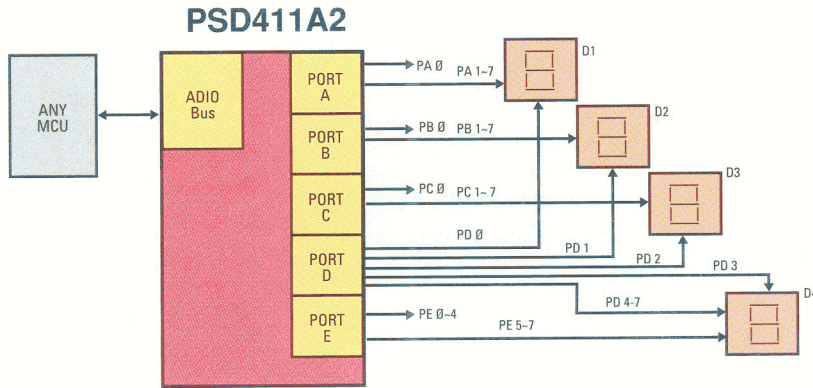
When used with inexpensive multiple sourced generic 8-bit MCUs, low cost PSDs provide manufacturing protection from sole-sourced MCUs and their uncertain supply issues.

## Eliminate Obsolete Designs

PSDs integrate a wide variety of system control functions into one device and save precious board space and power. For example, a powerful 16-bit bus 80C196 system can now be implemented with only two devices.



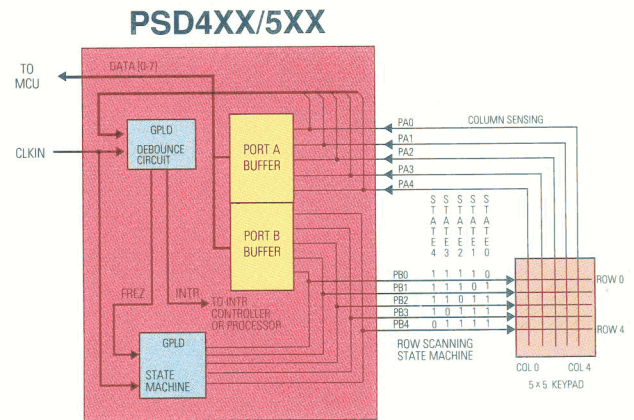
# Typical PSD Applications



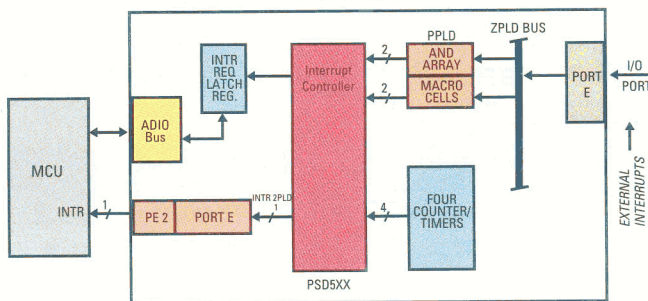
**PSD411A2 Directly Drives LED Display**  
 Sourcing 2 mA per segment and sinking 8 mA per digit, a PSD4XX/5XX can directly drive common-cathode LED displays. Drive options can be 100% duty cycle or 25% multiplexed duty cycle to reduce power by implementing a ring counter in the GPLD.

## Keypad Interface with Autoscanning and Debounce

The PSD4XX or PSD5XX can function as an interface to a keypad and perform both scanning and debounce by using the GPLD. An interrupt is provided to the MCU only when a key is pressed. For details, see Application Note 33 in the PSD Applications Handbook.



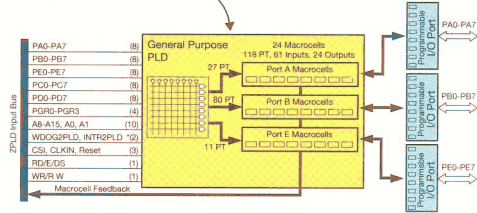
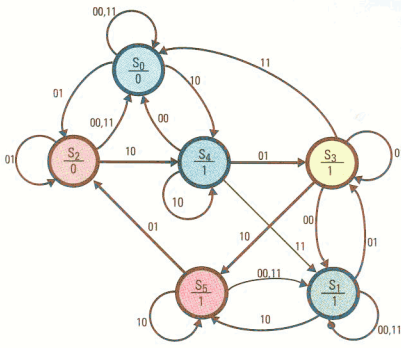
## PSD5XX



## External Interrupt Processing

The PSD5XX can simultaneously process a variety of regular or conditional external interrupts as well as internal Counter or Timer overflow interrupts. When an interrupt is signaled, the MCU reads the Interrupt Request Latch Register and the Interrupt Priority Status Register and then begins interrupt servicing.

### STATE MACHINE

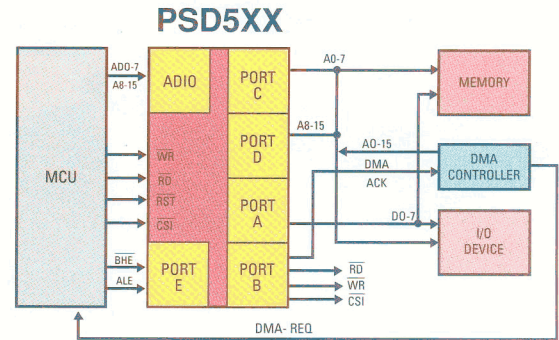


### State Machine Implemented on PSD4XX/5XX

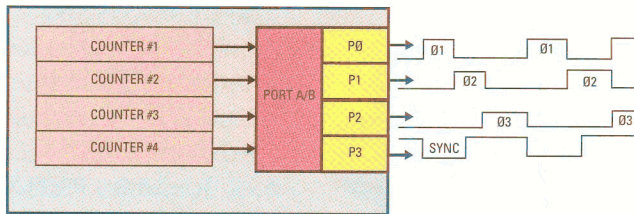
A state machine can be included in your next PSD design. The General Purpose PLD (GPLD) features 61 inputs plus clock and 118 output product terms. These outputs are fed to 24 macrocells, 8 each for Ports A, B, and E. User configurable, the macrocell outputs can be routed back to the ZPLD bus as macrocell feedback signals or used as outputs on the port pins.

### PSDs Facilitate DMA Transfers

The PSD Peripheral I/O mode can be used in DMA applications where the MCU does not tri-state the address/data bus. The MCU has access to any peripheral (memory or I/O device) through the PSD device. When a DMA request occurs, the MCU tri-states the address bus on PSD Ports C and D. The DMA controller then takes over the data and address buses after receiving acknowledgment from the MCU.



### PSD 5XX

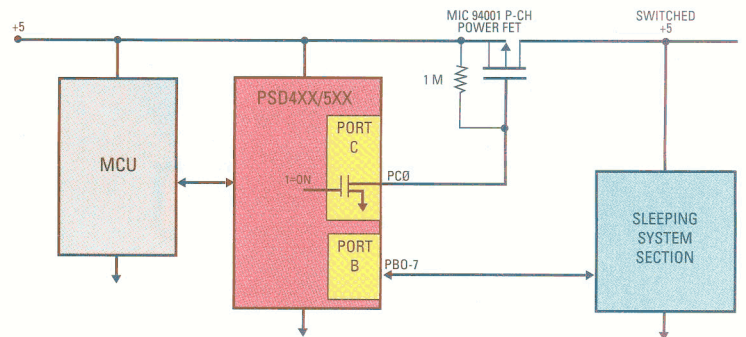


### Variable Duty-Cycle Waveform Outputs

By using its four Counter/Timers, the PSD5XX can simultaneously output four variable duty-cycle waveforms. The Pulse Width Modulated (PWM) signal duration can be updated every cycle. PWM drive signals for three-phase stepper motors can be easily implemented. For details, see Application Note 28, Appendix 6 in the PSD Applications Handbook.

### Power Switch

Any PSD4XX or PSD5XX Port C or D pin can be configured as an open-drain output. It can be used as a gate control on a P-channel high-side power FET that switches power to a system section such as a hard disk drive or cellular phone output stage that is turned off during periods of inactivity thus saving battery life. Any PSD I/O port defined as Standard MCU I/O or Address Out can be tri-stated by writing 00 to the Direction Register.



PSD devices are easy to configure and program. Next, review the PSDsoft Development Tools that can be used by almost anyone. →

# PSDsoft Development Tools



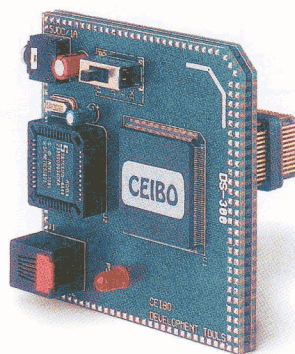
PSDsoft Development tools for PSD products operate on a 386/486 or Pentium® PC under Microsoft® Windows® 3.X or Windows 95®. WSI has seamlessly interfaced industry standard software design tools and applied them to the Windows framework to enable an easy "point and click" design process. By following a logical step-by-step design procedure, a PSD device can be quickly configured and programmed.

At \$99<sup>00</sup>, PSDsoft Lite is the lowest cost, entry-level PSD3XX development tool available.

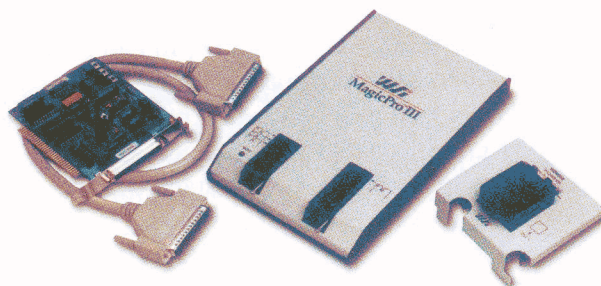
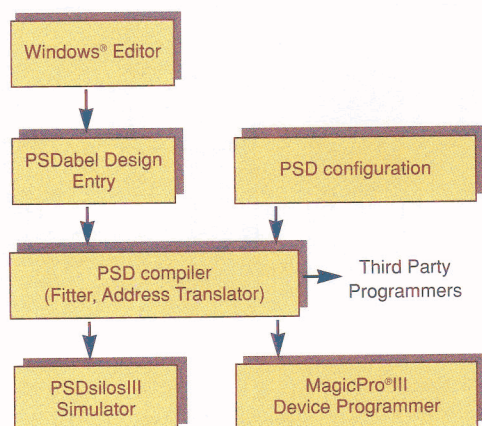


The five major software modules included in PSDsoft are:

- **PSDabel™**—the WSI Windows version of the popular Data I/O ABEL® logic design software.
- **PSD configuration**—specifies the MCU bus interface type and I/O port pin configurations.
- **PSD compiler**—the Fitter fits the logic and I/O functions into the PSD device of your choice. The Address Translator merges your code file and generates an object file which is used by the PSDsilosIII™ simulator or PSD programmer software. The file includes chip configuration and fuse map information along with EPROM program code.
- **PSDsilosIII™** simulator— the easy-to-use PSDsilosIII is the WSI version of Simucad's SILOS® simulation software. It provides full chip-level simulation and comprehensive node analysis.
- **PSD programmer**—used for programming a PSD device on the WSI MagicPro®III PC-compatible programmer.



The CEIBO DS-300 emulator is a complete Windows compatible PSD3XX software and hardware development tool. It enables PSD3XX prototype system operation from the keyboard.



**The MagicPro®III programmer** is an engineering development and prototyping tool and is used to program any PSD device. The programming menu in PSDsoft simplifies programming on the MagicPro®III.

# Getting started with PSD Products is easy

Just follow these five easy steps:

## Step 1:

Review the PSDsoft Demo Disk. It illustrates the PSD design procedure and includes actual PSDsoft Windows screens.



## Step 2:

Call your local WSI Sales Representative listed on the back of this brochure and get free copies of the PSD Products and Applications data books. Application Note 31 entitled "PSD5XX Design Tutorial" presents a detailed step-by-step training procedure for getting started with PSD products. WSI's field applications engineers can help you choose the right PSD device to improve your system.

Visit our PSD Design Center on the web at [www.wsipsd.com](http://www.wsipsd.com)



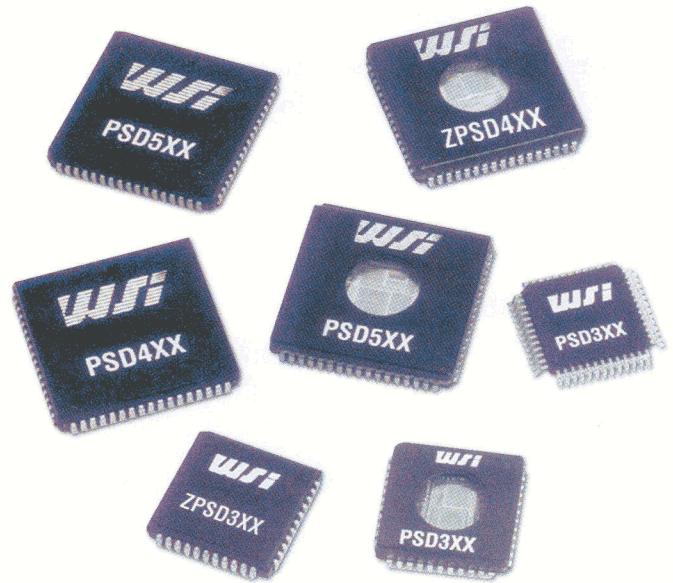
## Step 3:

Purchase a low-cost PSDsoft development system and a MagicPro III programming unit. These tools provide everything you need for rapid completion of your PSD design. If you already have an industry standard programmer, you may be ready to program PSD devices now.



## Step 4:

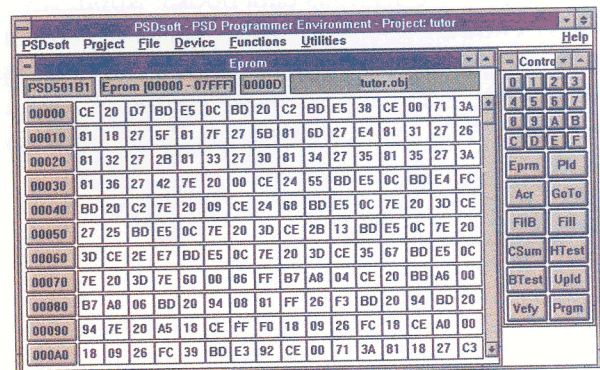
Install PSDsoft on your PC and proceed with your design. The CEIBO PSD 3XX Emulator and Application Note 32 ("Use a ROM Emulator...") are helpful design tools. Get free PSD product samples for prototyping from your local WSI Sales Representative. Field application engineers and factory support personnel are available for any questions or problems. The WSI Bulletin Board (Tel. 800-868-1010 or 510-498-1002) is also available. Contact CEIBO at tel: 314-830-4084 or e.mail 100274.2131@compuserve.com



## Step 5

Program a sample PSD device and plug it into your prototype. Something doesn't work right? No problem! Simply erase the part, adjust your software or the PSD device configuration with PSDsoft and re-program the unit. Plug it back in and try again until everything works right.

That's all there is to it!



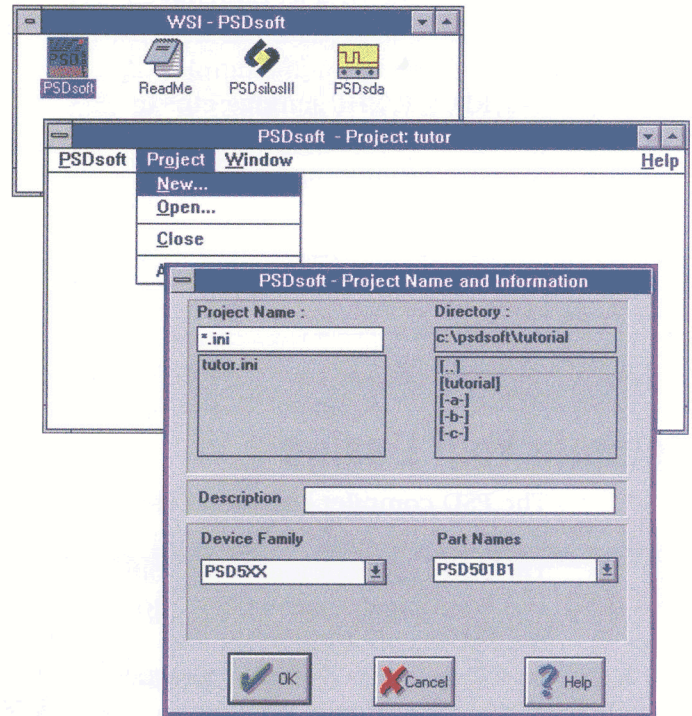
# A Typical PSD Design Sequence

Windows based PSDsoft is very easy to use. The following is a step-by-step procedure for configuring and programming a PSD device for your embedded control design.

## Project Management

**1** After loading PSDsoft, double click the ReadMe icon on the PSDsoft window. ReadMe contains important information you should know before proceeding with your design.

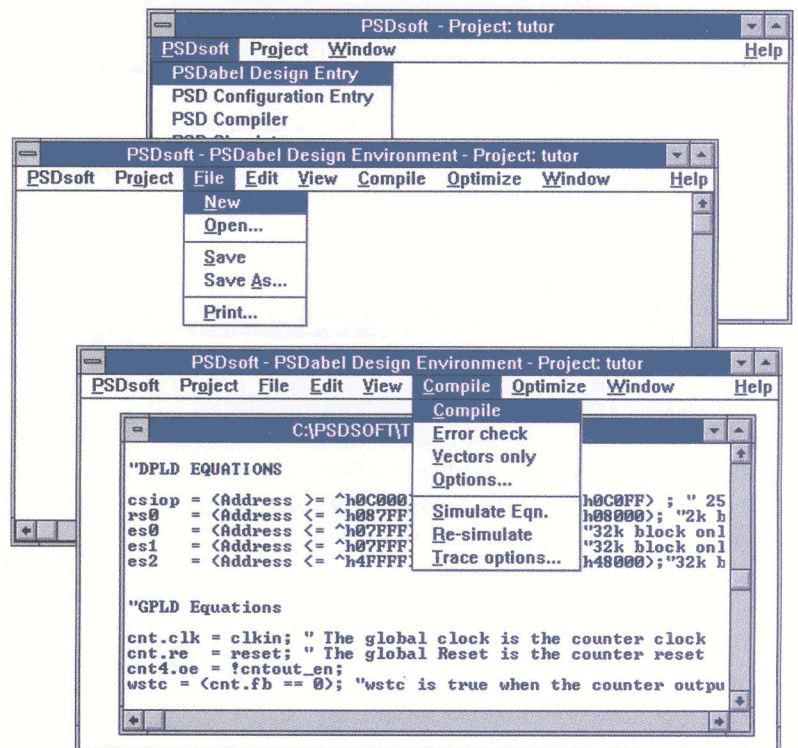
Next, assign a name to your project so the PSDsoft Project Management software can transfer the working directory and pertinent information to other functional modules. Also, enter the unique PSD part number you plan to work with.



## Enter Your Design Source File

**2** PSDabel is the design entry tool used to define the PLDs and some I/O constructs. Pull down the PSDsoft menu and choose PSDabel Design Entry. PSDabel is WSI's version of Data I/O's popular ABEL-HDL design software. It includes all the ABEL functions required to compile, optimize, and simulate PLD source files written in ABEL Hardware Description Language (PSDabel-HDL). Refer to the PSDsoft User Manuals if you have any questions and then complete the source files for your design.

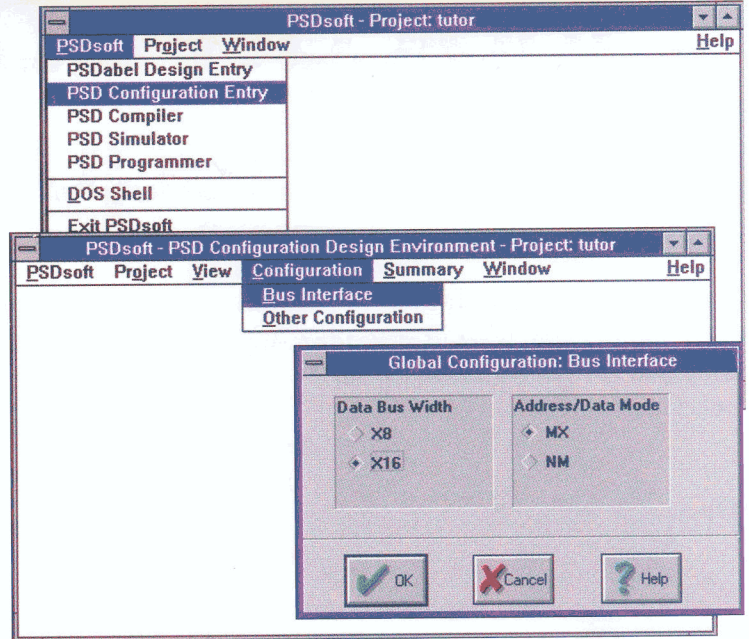
Next, compile your source file. The Compile Menu includes items used by the PSDabel simulator. After compiling, you can simulate your logic design equations and view the results in a PSDabel waveform file.



## Configure Your Design

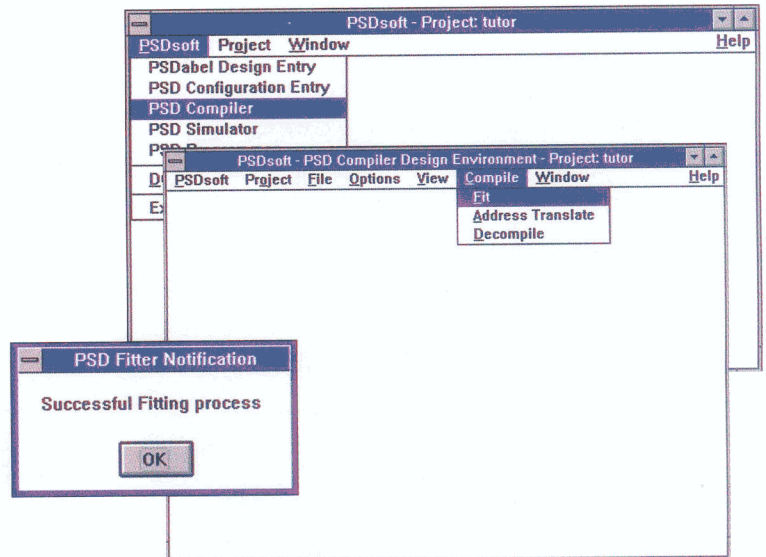
**3** After simulating your source file and verifying all logic functions are correct, the next step is to configure your PSD device.

A PSD device has a configurable MCU bus interface and is able to interface to virtually any microcontroller. PSD configuration software enables you to select the bus configuration of your choice: data bus width (8- or 16-bit), multiplexed or non-multiplexed, ALE/AS polarity, RD/WR/BHE settings, etc. In addition, you can select the configuration of the PSD5XX's counter/timers and interrupt controller.

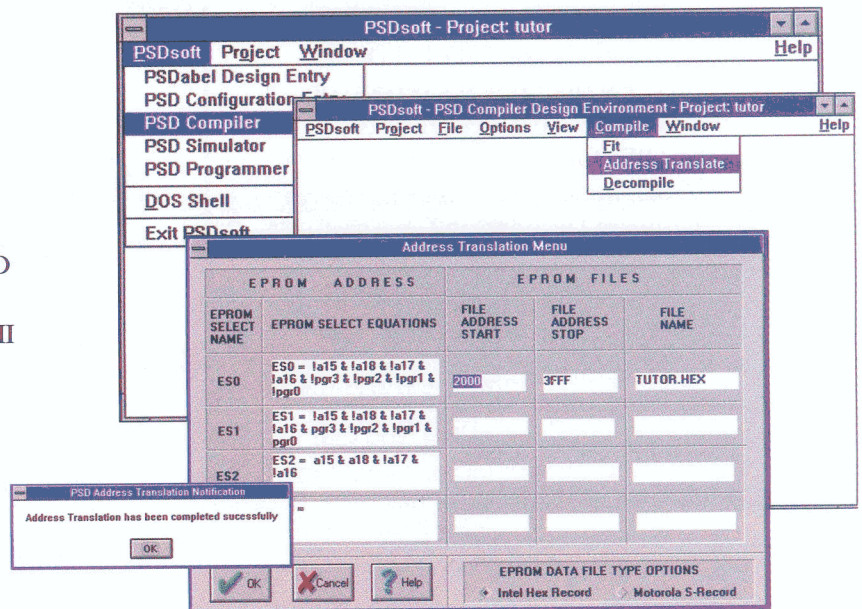


## Compile Your Design

**4** The PSD compiler consists of the Fitter and the Address Translator. The Fitter fits your logic functions and configuration selections into the PLD arrays. If there are any problems, you will be advised via an error message file regarding which logic function is causing the fitting problem. You can then modify the source file accordingly.



The **Address Translator** combines the fuse map file with your EPROM code as well as PSD configuration information. It generates a file used by the PSDsilosIII simulator or MagicPro III programmer to program your PSD devices.

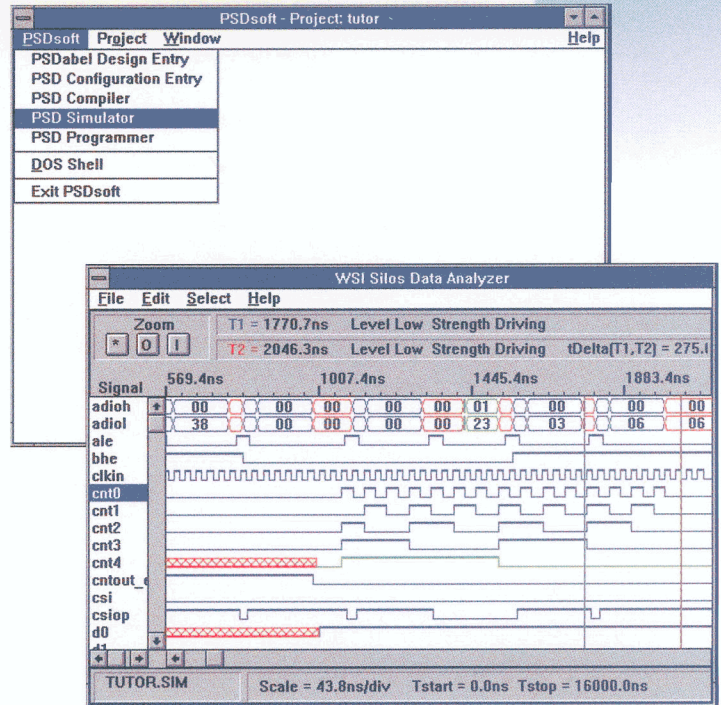


## Simulate Your Design

**5** You might prefer to simulate your PSD design at the chip-level. For this purpose, WSI supplies PSDsilosIII which can model your design using the Verilog Language. Well over 100 internal and external PSD device nodes are available for tracing and design timing analysis after simulation.

The PSDsilosIII Data Analyzer enables you to view a list of signals selected from the Node List. You can save any grouping for later inspection or comparison. Waveform signals may be moved on the display and the window can be scrolled up, down, zoomed in, zoomed out, etc. Two time marker cursors specify the time range to be displayed. Signals on the display consist of various colors depending on their type and signal strength.

By selecting any group of signals for display, you can quickly determine timing correlation in such areas as bus activity, internal function selection versus address value, counting/timing relationships relative to logic events, etc. This powerful tool greatly shortens system design time.

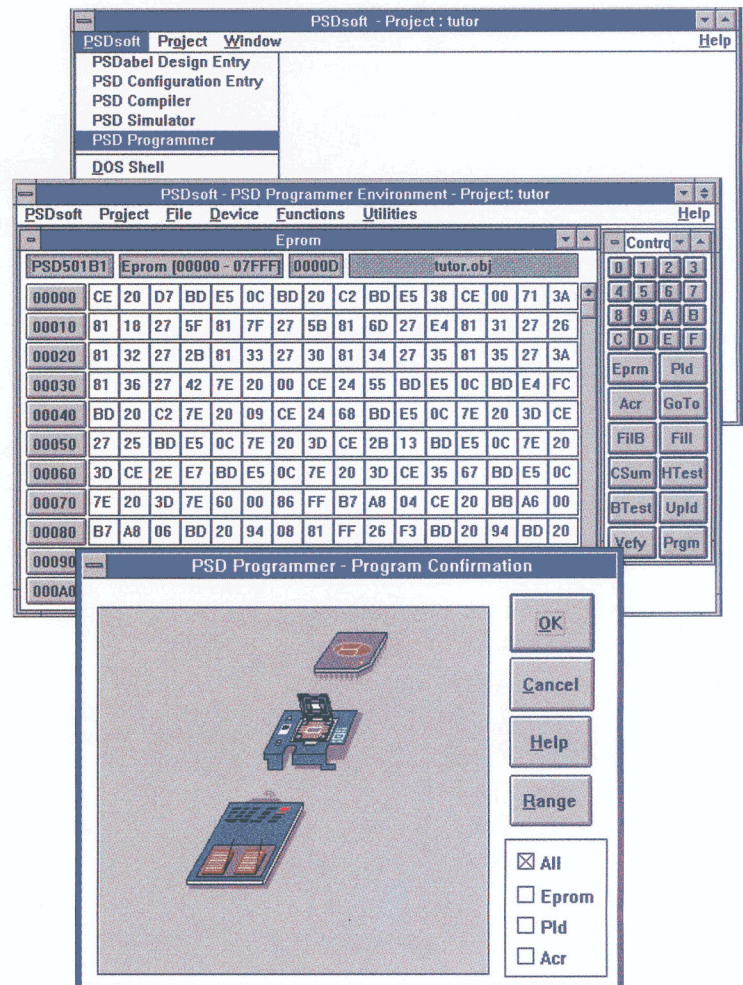


## Program Your PSD Device

**6** After verifying your design through simulation, the last step is to program the PSD device. When using PSD programmer software, the contents of your object file can be displayed and edited for any last minute code changes.

Then, by using the WSI MagicPro III programmer, you can Blank Test, Upload Data, or Program and Verify your target PSD device. During programming, the MagicPro III performs a blank check, programs each location, and automatically executes a program verify.

A certified industry standard programmer from anyone of the vendors in WSI's Preferred PSD Programmer series may also be used.



**That's all there is to it! Now plug in your newly designed PSD peripheral into your prototype and apply power.**

# WSI Worldwide Sales, Service and Technical Support

## WSI USA REPS

### ALABAMA

Rep. Inc.  
Tel: (205) 881-9270  
Fax: (205) 882-6692

### ARIZONA

Summit Sales  
Tel: (602) 998-4850  
Fax: (602) 998-5274

### CALIFORNIA

Bager Electronics, Inc.  
Tel: (714) 957-3367  
Fax: (714) 546-2654

Tel: (818) 712-0011  
Fax: (818) 712-0160

Earle Assoc., Inc.  
Tel: (619) 278-5441  
Fax: (619) 278-5443

I Squared, Inc.  
Tel: (408) 988-3400  
Fax: (408) 988-2079  
Tel: (916) 989-0843  
Fax: (916) 989-2841

### CANADA

Intelatech, Inc.  
Tel: (905) 629-0082  
Fax: (905) 629-1795

### COLORADO

Waugaman Associates, Inc.  
Tel: (303) 423-1020  
Fax: (303) 467-3095

### CONNECTICUT

Advanced Tech Sales  
Tel: (508) 664-0888  
Fax: (508) 664-5503

### FLORIDA

QXi of Florida, Inc.  
Tel: (954) 341-1440  
Fax: (954) 341-1430  
Tel: (407) 576-3445  
Fax: (407) 576-2601  
Tel: (813) 576-3445  
Fax: (813) 576-2601

### GEORGIA

Rep. Inc.  
Tel: (770) 938-4358  
Fax: (770) 938-0194

### IDAHO

Badger Electronics  
Tel: (801) 582-0501  
Fax: (801) 582-1850

### ILLINOIS

Victory Sales  
Tel: (847) 490-0300  
Fax: (847) 490-1499

### INDIANA

Victory Sales  
Tel: (317) 581-0880  
Fax: (317) 581-0882

### IOWA

Gassner & Clark Co.  
Tel: (319) 393-5763  
Fax: (319) 393-5799

### KANSAS/NEBRASKA

Rush & West Associates  
Tel: (913) 764-2700  
Fax: (913) 764-0096

### KENTUCKY

Victory Sales  
Tel: (513) 436-1222  
Fax: (513) 436-1224

### MARYLAND/VIRGINIA

Robert Electronic Sales  
Tel: (410) 995-1900  
Fax: (410) 964-3364

### MASSACHUSETTS

Advanced Tech Sales, Inc.  
Tel: (508) 664-0888  
Fax: (508) 664-5503

### MICHIGAN

Victory Sales  
Tel: (313) 432-3145  
Fax: (313) 432-3146

### MINNESOTA

OHMS Technology, Inc.  
Tel: (612) 932-2920  
Fax: (612) 932-2918

### MISSOURI

Rush & West Associates  
Tel: (314) 965-3322  
Fax: (314) 965-3529

### NEW JERSEY

Strategic Sales, Inc.  
Tel: (201) 842-8960  
Fax: (201) 842-0908

BGR WYCK  
Tel: (609) 727-1070  
Fax: (609) 727-9633

### NEW MEXICO

S & S Technologies  
Tel: (602) 438-7424  
Fax: (602) 414-1125

### NEW YORK

Strategic Sales, Inc.  
Tel: (201) 842-8960  
Fax: (201) 842-0906

Tri-Tech Electronics, Inc.  
Tel: (716) 385-6500  
Fax: (716) 385-7655

Tel: (607) 722-3580  
Fax: (607) 722-3774

### NORTH CAROLINA

Rep. Inc.  
Tel: (919) 469-9997  
Fax: (919) 481-3879

### OHIO

Victory Sales  
Tel: (216) 498-7570  
Fax: (216) 498-7574

### OREGON

Victory Sales  
Tel: (513) 436-1222  
Fax: (513) 436-1224

### OKLAHOMA

AXXIS Technology Mktng.  
Tel: (972) 491-3577  
Fax: (972) 491-2508

### OREGON

Electra Technical Sales  
Tel: (503) 643-5074  
Fax: (503) 526-2055

### PENNSYLVANIA

Victory Sales  
Tel: (216) 498-7570  
Fax: (216) 498-7574

### BGR WYCK

Tel: (609) 983-1020  
Fax: (609) 983-1879

### PUERTO RICO

QXi of Florida, Inc.  
Tel: (305) 978-0120  
Fax: (305) 972-1408

### TENNESSEE

Rep. Inc.  
Tel: (423) 475-9012  
Fax: (423) 475-6340

### TEXAS

AXXIS Technology Mktng.  
Tel: (512) 320-9130  
Fax: (512) 320-5730

Tel: (972) 491-3577  
Fax: (972) 491-2508

Tel: (713) 338-9901  
Fax: (713) 332-0610

### UTAH

Bager Electronics  
Tel: (801) 582-0501  
Fax: (801) 582-1850

### WASHINGTON

Electra Technical Sales  
Tel: (206) 821-7442  
Fax: (206) 821-7289

### WISCONSIN

Victory Sales  
Tel: (414) 789-5770  
Fax: (414) 789-5760

OHMS Technology, Inc.  
Tel: (612) 932-2920  
Fax: (612) 932-2918

## DISTRIBUTORS

Arrow Electronics  
Avnet Electronics  
Marsh Electronics  
Port Electronics  
Time Electronics  
Vantage Components  
Wyle Laboratories  
Zeus Electronics

## WSI WORLDWIDE

### AUSTRALIA

Zatek Components  
Tel: 61-2-9744-5711  
Fax: 61-2-9744-5527

Tel: 61-3-9574-9644  
Fax: 61-3-9574-9661

### BELGIUM, LUX

Alcom Electronics nv/sa  
Tel: 32-3-458-3033  
Fax: 32-3-458-3126

### BRAZIL/ARGENTINA

Colgil, Inc.  
Tel: 55-11-663285  
Fax: 55-11-663285

### CHINA

Comex Technology  
Tel: (86-10)849-9430/8888  
Fax: (86-10)849-9430

Tel: (86-811)531-5258  
Fax: (86-811)531-5258

Tel: (86-20)380-7307/5688  
Fax: (86-20)380-7307

Tel: (86-25)449-1384  
Fax: (86-25)449-1384

Microlink Intl Co.  
Tel: (602)276-7808  
Fax: (602)276-8211

Wuhan Liyuan  
Computer Ltd.  
Tel: 86-27-7802986  
Fax: 86-27-7802985

### DENMARK

Jakob Hatteland A/S  
Tel: (45) 42-571000  
Fax: (45) 42-166199

### ENGLAND

Micro Call, Ltd.  
Tel: 44-184-426-1939  
Fax: 44-184-426-2998

Silicon Concepts, Ltd.  
Tel: 44-1428-751-617  
Fax: 44-1428-751-603

### EASTERN EUROPE

Elatec Vertriebs.  
Tel: 49-89-462-3070  
Fax: 49-89-460-2403

### FINLAND

Avnet Nortec OY  
Tel: 358-0613181  
Fax: 358-06922326

### FRANCE

Axess Technology  
Tel: 33 (1) 49-78-94-94  
Fax: 33 (1) 49-78-03-24

### Microel

Tel: 33 (1) 69-07-08-24  
Fax: 33 (1) 69-07-17-23

### GERMANY

Jermyn GmbH  
Tel: (06) 431-5080  
Fax: (06) 431-508289

### Scantec GmbH

Tel: 49 (089) 899-1430  
Fax: 49 (089) 857-6574

### Topas Electronic GmbH

Tel: 49 (0511) 968640  
Fax: 49 (0511) 9686464

### HONG KONG

Comex Technology Ltd.  
Tel: 852-2735-0325  
Fax: 852-2730-7538

### INDIA/PAKISTAN

Pamir Electronics Corp.  
Tel: 610-594-8337  
Fax: 610-594-8559

### ISRAEL

Star-Tronics, Ltd.  
Tel: 972-3-6960148  
Fax: 972-3-6960255

### ITALY

Compres SPA  
Tel: 39-3625781  
Fax: 39-362553967

Silverstar  
Tel: 39 2661251  
Fax: 39 266101359

### JAPAN

Internix, Inc.  
Tel: 813-3-369-1105  
Fax: 813-3-363-8486

Kyocera Corporation  
Tel: 813-3-708-3111  
Fax: 813-3-708-3372

Nippon Imex Corporation  
Tel: 813-3-321-8000  
Fax: 813-3-325-0021

### KOREA

Woo Young Tech Co., Ltd.  
Tel: 82-2-369-7099  
Fax: 82-2-369-7091

### NETHERLANDS

Alcom Electronics  
Tel: 31-10-451-9533  
Fax: 31-10-458-6482

### NEW ZEALAND

Apex Electronics  
Tel: 644-3853404  
Fax: 644-3853483

### NORWAY

Henaco A/S  
Tel: 47-22-16-21-10  
Fax: 47-22-25-77-80

### REPUBLIC OF SOUTH AFRICA

Components & System Design  
Tel/Fax: 2711-391-3062

### SINGAPORE

Technology Distribution(s) Pte, Ltd.  
Tel: 65-299-7811  
Fax: 65-294-1518

### SPAIN, PORTUGAL

Matrix Electronica  
Tel: 34.1.3272778  
Fax: 34.1.3041534

### SWEDEN

DipCom Electronics  
Tel: 46-8-7522480  
Fax: 46-8-7513649

### SWITZERLAND

Elbatex  
Tel: (41) 56-43-75-11-11  
Fax: (41) 56-26-14-86

### Laser & Electronic Equipment

Tel: 41-1-4223330  
Fax: 41-1-4223458

### TAIWAN

Ally, Inc.  
Tel: 886-2-768-6399  
Fax: 886-2-768-6390

Visit our WebSite at: <http://www.wsipsd.com>

### Midwest

Hoffman Estates, IL  
Tel: (847) 215-2560  
Fax: (847) 215-2702

### West

Irvine, CA  
Tel: (714) 753-1180  
Fax: (714) 753-1179

### Northeast

Trevose, PA  
Tel: (215) 638-9617  
Fax: (215) 638-7326

### Southeast

Dallas, TX  
Tel: (972) 418-2970  
Fax: (972) 418-2971

### EUROPE SALES

WSI - France  
2 voie LA CARDON  
91126 Palaiseau  
CEDEX, France  
Tel: 33 (1) 69-32-01-20  
Fax: 33 (1) 69-32-02-19

### ASIA SALES

WSI Korea  
Tel: 82-2-751-1281/2  
Fax: 82-2-761-1283



### Corporate Headquarters

47280 Kato Road  
Fremont, California 94538-7333  
Tel: 510-656-5400 Fax: 510-657-5916  
800-TEAM-WSI (800)-832-6974  
In California 800-562-6363

WSI -

# Les périphériques programmables PSD3XX de WSI

*Outre l'unité centrale, les cartes à microprocesseur rassemblent traditionnellement de multiples circuits périphériques et des mémoires.*

*Les microcontrôleurs intègrent en principe tout cela dans un seul et même boîtier, mais leurs ressources peuvent se révéler insuffisantes pour bien des projets d'une certaine ampleur : il faut alors leur ajouter de la mémoire ou même des périphériques, ce qui ramène au cas précédent ! L'innovation majeure introduite par WSI avec sa famille de "périphériques programmables" est de réunir dans un unique boîtier, des périphériques d'entrée-sortie et une généreuse quantité de mémoire RAM et EPROM.*

La logique interne de configuration et d'adressage étant programmable tout comme un simple PLD, quelques références suffisent pour assurer la compatibilité avec pratiquement n'importe quel microprocesseur ou microcontrôleur.

Des solutions à deux boîtiers seulement sont donc envisageables, même pour des applications d'un haut degré de complexité.

## TOUT SAUF L'UNITE CENTRALE

L'architecture typique d'un système microprogrammé est bien connue : une unité centrale, de la mémoire ROM et RAM, et différents périphériques. Le tout relié par des bus de données, d'adresse et de contrôle munis de circuits de décodage ou de sélection.

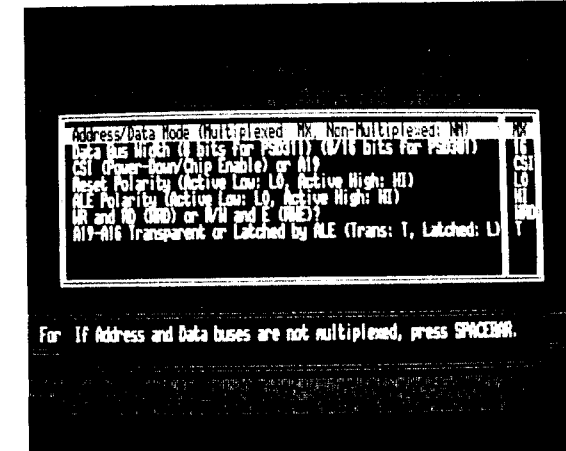
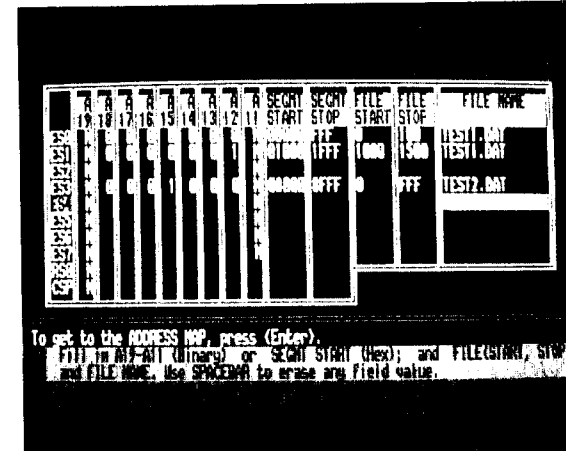
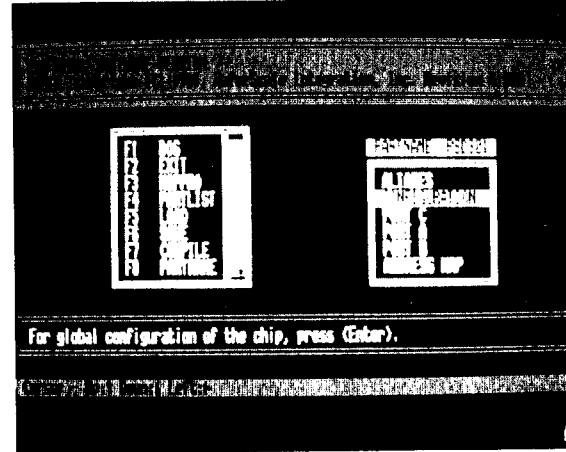
Si l'unité centrale est un simple microprocesseur, il est habituel de lui associer au moins deux boîtiers de mémoire, un ou deux boîtiers d'entrée-sortie, et soit un ou deux PAL de décodage, soit cinq ou six composants CMOS ou TTL.

L'utilisation de microcontrôleurs permet de concentrer pratiquement tout cela dans un unique boîtier, du moins tant que l'on n'a pas besoin de trop de capacité mémoire ou d'un trop grand nombre de lignes d'entrée-sortie. Si l'on doit ajouter de la mémoire externe, il faut la relier au microcontrôleur par des bus qui monopolisent alors une bonne partie des lignes d'entrée-sortie disponibles.

Leur éventuelle reconstitution nécessite alors plusieurs boîtiers, auxquels il faut ajouter le démultiplexeur habituellement utilisé pour interfacer la mémoire (voir l'exemple de la figure 1).

A vrai dire, la solution strictement "monochip" ne concerne qu'une minorité d'applications relativement simples.

Or, il devient de plus en plus souvent nécessaire de développer des systèmes complexes mais miniaturisés et à faible consommation, et ce dans des délais toujours plus courts : des



exemples typiques sont les radiotéléphones cellulaires, les contrôleurs de disques durs, les modems, l'instrumentation, les périphériques d'ordinateurs, et d'une façon générale beaucoup

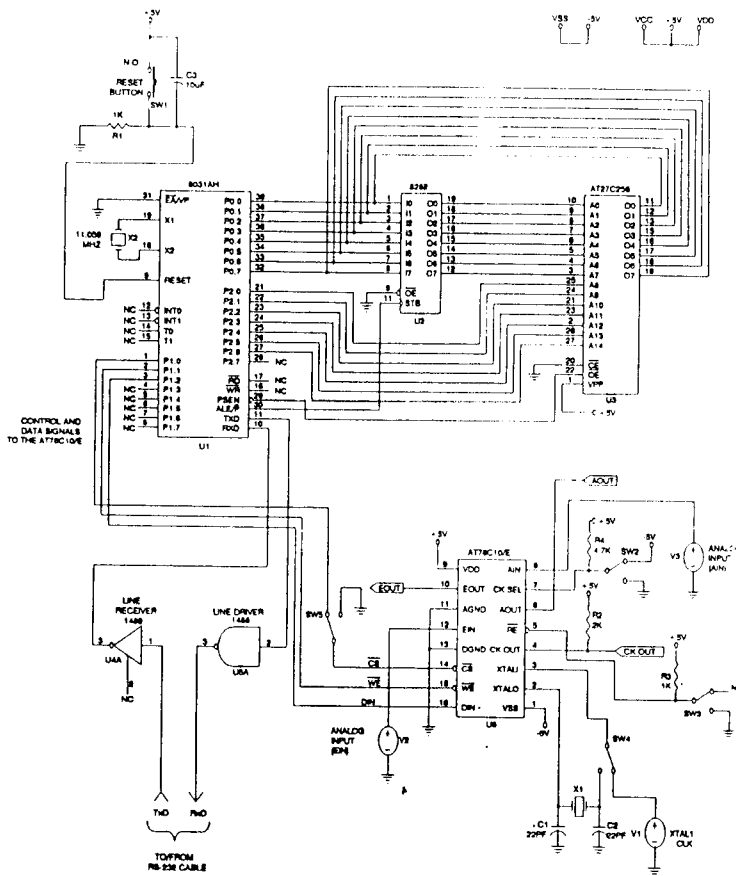


Figure 1

de systèmes industriels, automobiles, médicaux ou militaires. Les périphériques programmables de WSI (Wafer Scale Integration) permettent de concilier tous ces impératifs, et représentent donc une innovation importante pour le développeur de systèmes de moyenne ou grande complexité.

Les composants de la famille PSD 3XX rassemblent en effet dans un seul boîtier (céramique à fenêtre ou OTP) tout ce qu'il faut pour ajouter des ports d'entrée-sortie, 256 à 1024 k-bits d'EPROM, et 16 k-bits de RAM statique à un microprocesseur ou microcontrôleur pratiquement quelconque : 6805, 68HC11, 68000/10/20, 8031/8051, 8096/8098, 80186/88, 80196/98, Z80, Z8, etc. ou même à un DSP.

Certaines références offrent par ailleurs une possibilité d'extension de l'espace adressable d'origine d'un microcontrôleur par une technique de pagination : jusqu'à 16 pages d'un mégabit !

Mais comment six modèles seulement peuvent-ils offrir une telle souplesse ?

Tout simplement parce que la logique interne des PSD 3XX est bâtie autour de réseaux logiques programmables comparables à des PAL : un logiciel spécial permet à l'utilisateur de décrire son

système à l'aide du premier PC venu, puis un programmeur approprié personnalise le composant en conséquence.

Un atout majeur de cette démarche est que les PSD 3XX disposent d'un "bit de sécurité" comme n'importe quel PAL : une fois programmé, il interdit toute relecture des données de configuration et donc toute tentative de "reverse engineering".

Même le contenu de l'EPROM, pourtant accessible librement par l'unité centrale, peut être considéré comme protégé : en effet, comme la carte de la mémoire et des entrées-sorties ne peut être consultée, il n'est pratiquement pas possible de savoir à quoi correspondent les octets que l'on pourrait arriver à lire par un moyen ou par un autre...

### Le PSD 301

La famille PSD 3XX se compose actuellement de six références, qui ne diffèrent pas très profondément les unes des autres : elles sont en fait chacune plus ou moins optimisées pour certaines catégories de processeurs, et incorporent éventuellement des perfectionnements supplémentaires.

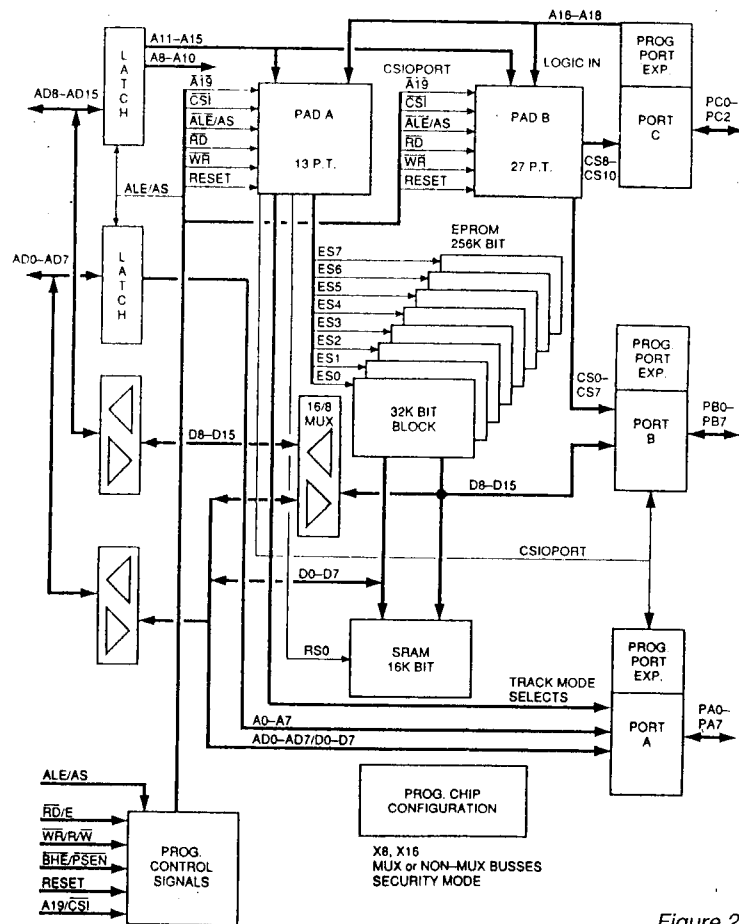


Figure 2

Le PSD 301 est le premier en date de ces composants mais reste assez représentatif de l'ensemble de la gamme, qui comprend les références suivantes : PSD 301, PSD 311, PSD 302, PSD 312, PSD 303 et PSD 313.

Le PSD 301, dont la **figure 2** reproduit le schéma synoptique, offre 19 lignes d'entrée-sortie configurables individuellement, et pouvant donc être utilisées indifféremment comme :

- extension des entrées-sorties du microcontrôleur hôte,
- entrées-sorties du décodeur d'adresses programmable,
- sorties d'adresses verrouillées,
- lignes à drain ouvert ou CMOS.

Deux réseaux logiques programmables dits "Programmable Address Decoders" (PAD A et PAD B) offrent un total de 40 termes de produits et jusqu'à 16 entrées et 24 sorties. Ils remplacent la totalité de la logique d'appoint habituelle, et offrent une capacité de décodage d'adresses d'un mégabit.

Des verrous sont prévus pour la gestion de bus de données et d'adresses multiplexés, mais le circuit reste évidemment compatible avec les bus non multiplexés.

Dans les deux cas, les bus à 8 ou 16 bits sont supportés. La polarité des signaux ALE et RESET est librement programmable, tandis que le bus de contrôle peut être du type RD barre/WR barre ou R/W barre/E.

Une broche BHE est prévue pour le mode 16 bits, ainsi qu'une broche PSEN pour les utilisateurs de 8051.

256 k-bits d'EPROM effaçable aux UV (sauf boîtiers OTP) sont disponibles, sous la forme 32 k x 8 ou 16 k x 16.

Le tout est divisé en huit blocs pouvant être répartis librement dans l'espace adressable : cela accroît la souplesse du système et contribue à la protection du contenu de l'EPROM contre les tentatives de lecture ou de copie.

S'y ajoutent 16 k-bits de RAM statique 120 ns, organisée en 2 k x 8 ou 1 k x 16.

Enfin, des facilités sont prévues pour le partage de ressources entre plusieurs microcontrôleurs ou avec un processeur hôte.

Le cœur du PSD 301 est évidemment son réseau logique programmable, dont la **figure 3** dévoile l'organisation. On retrouve l'architecture classique des PAL (réseau programmable de portes ET suivi d'un réseau fixe de portes OU).

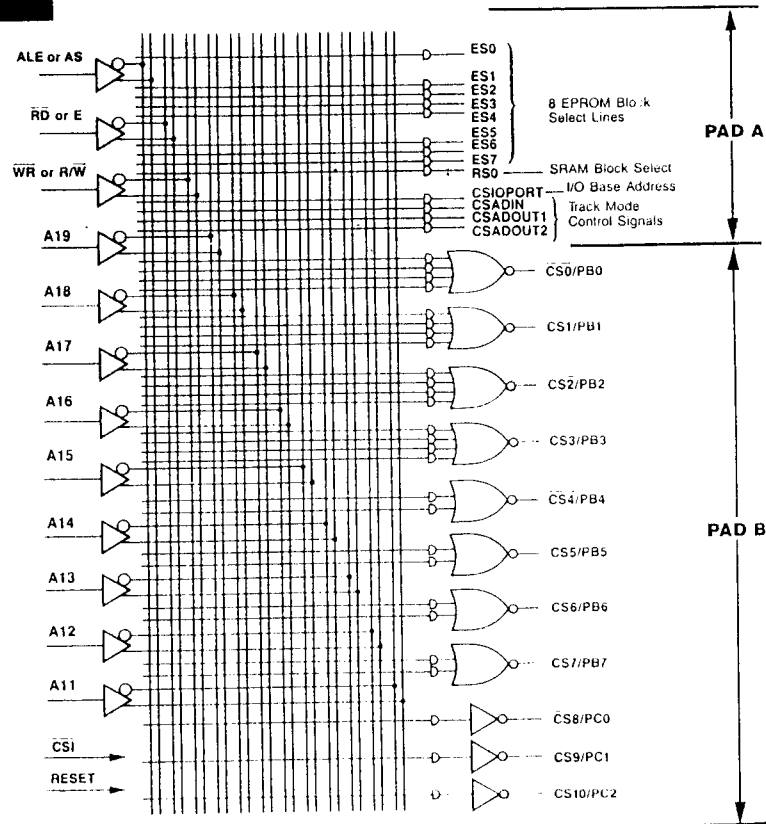


Figure 3

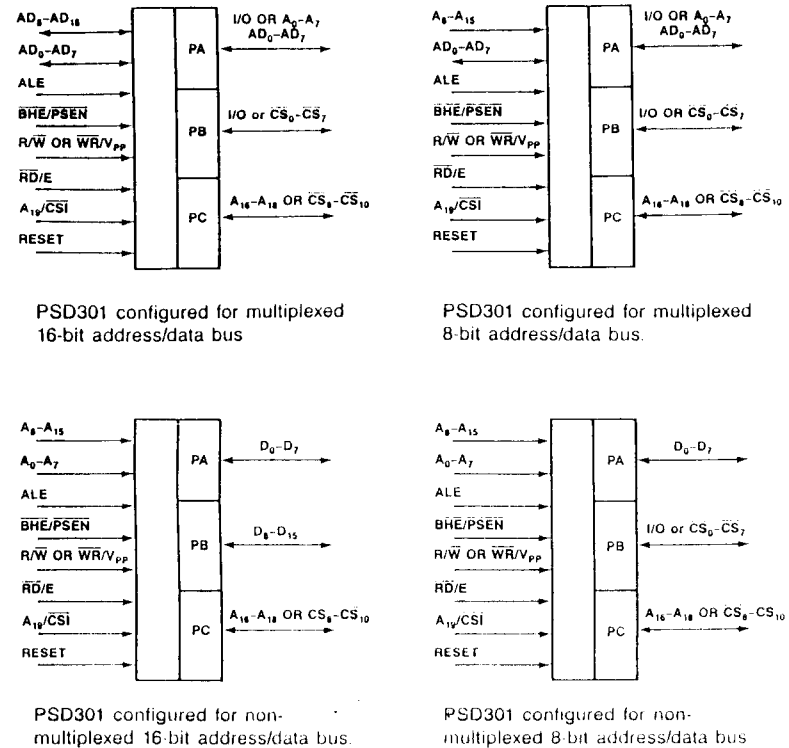


Figure 4

Par rapport à un PAL courant comme le 16L8, par exemple, la matrice programmable est plus petite, mais davantage optimisée : il en résulte un nombre d'entrées et de sorties supérieur, réparti d'ailleurs en deux groupes distincts (PAD A et PAD B) dont chacun remplace pratiquement l'équivalent d'un PAL classiquement sous-employé.

C'est par simple programmation de ce réseau logique que l'on peut adapter très exactement l'architecture du PSD 301 à celle du système final, et notamment aux caractéristiques de ses bus comme le montre le **figure 4** : 8 ou 16 bits, multiplexés ou non. Cela suppose une structure assez complexe des ports d'entrée-sortie, directement placés

sous le contrôle du réseau programmable : la **figure 5** décrit l'agencement du port A, la **figure 6** celui du port B, et la **figure 7** celui du port C.

Quarante quatre broches sont nécessaires pour assurer la liaison du PSD 301 avec le "monde extérieur", processeur compris.

Plusieurs boîtiers sont disponibles, avec ou sans fenêtre pour l'effacement aux UV : un LCC céramique ou plastique à 44 broches (**figures 8 et 9**), un QFP

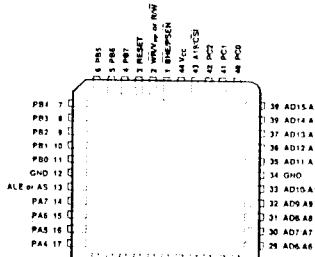


Figure 8

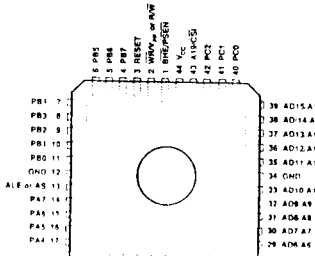


Figure 9

plastique à 52 broches (**figure 10**) et un PGA céramique à 44 broches (**figure 11**).

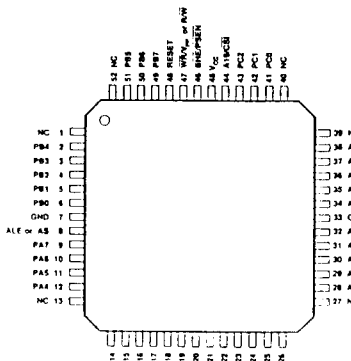


Figure 10

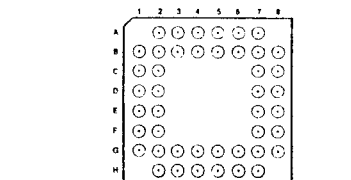


Figure 11

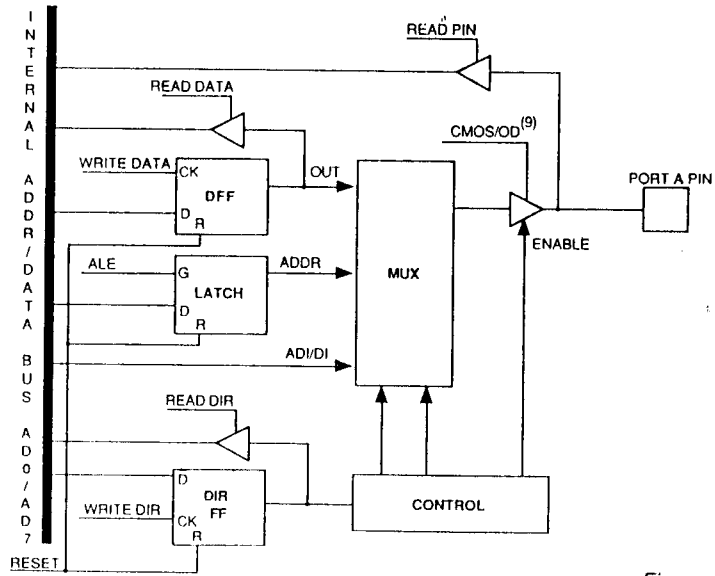


Figure 5

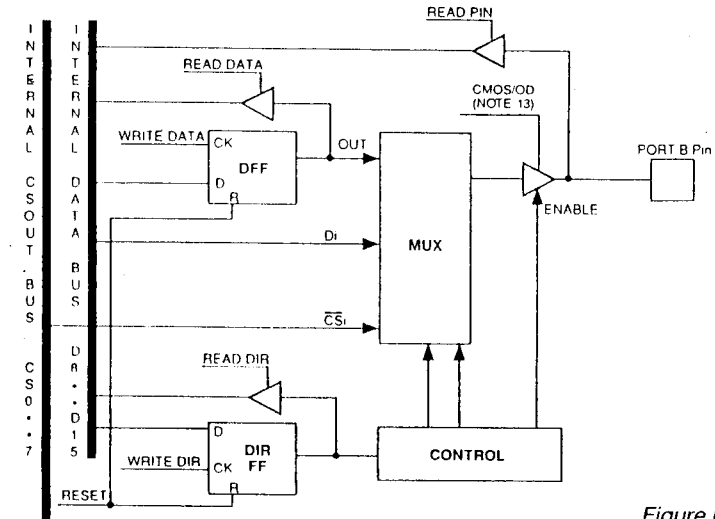


Figure 6

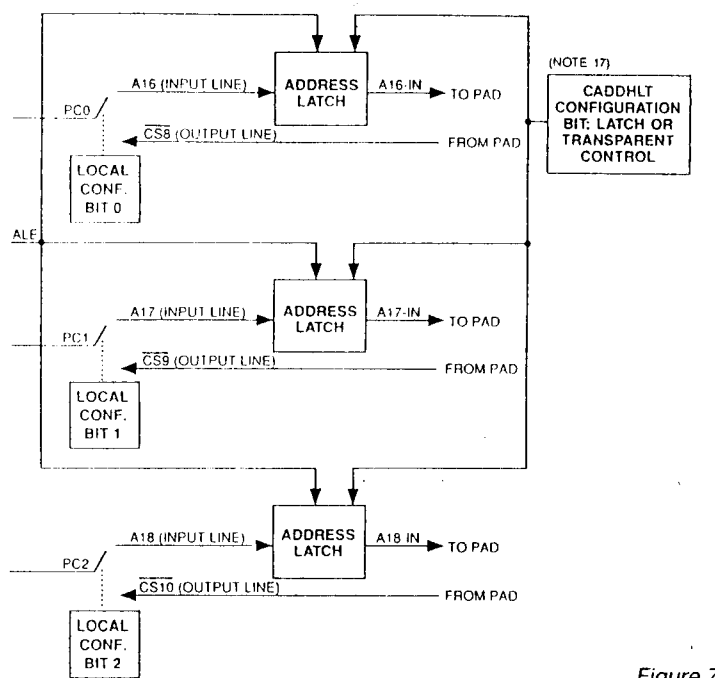


Figure 7

La réalisation de ces deux systèmes est décrite dans les schémas suivants, qui sont à considérer comme des exemples de réalisation PSD 3XX.

La réalisation d'adaptateurs de ces composants est faite par définition à des choses applicables à un cas particulier. Plus souvent au niveau matériel, on s'occupe d'adapter le processeur, son programme, son matériel, et le "vendeur" lui-même, mais certainement au niveau de la programmation du PSD 3XX. Les exemples de schémas qui sont présentés dans ces pages visent à servir de guide aux PSD 3XX à différents niveaux matériels et peuvent être très commodes à la réalisation d'applications. À cet égard, les schémas complètent donc un système par lequel, respectivement, de l'extérieur et du niveau programme de son PSD 3XX.

Les exemples sont par deux exemples particulièrement typiques d'adaptation des PSD 3XX sur des microcontrôleurs très populaires : le 80C31 à la figure 13, et le 68HC11 à la figure 14.

Dans un cas comme dans l'autre, les deux ports d'entrée-sortie servant à la communication avec le PSD 3XX sont reconstruits, tandis que ce périphérique programme ses propres données en mémoire RAM et EPROM protégée.

Tout cela moyennant un montage bien plus compliqué ou encombrant que s'il s'agissait d'ajouter une simple EPROM ! Passons maintenant aux microprocesseurs, dépourvus par définition de tout port d'entrée-sortie. Les PSD 3XX s'interfacent

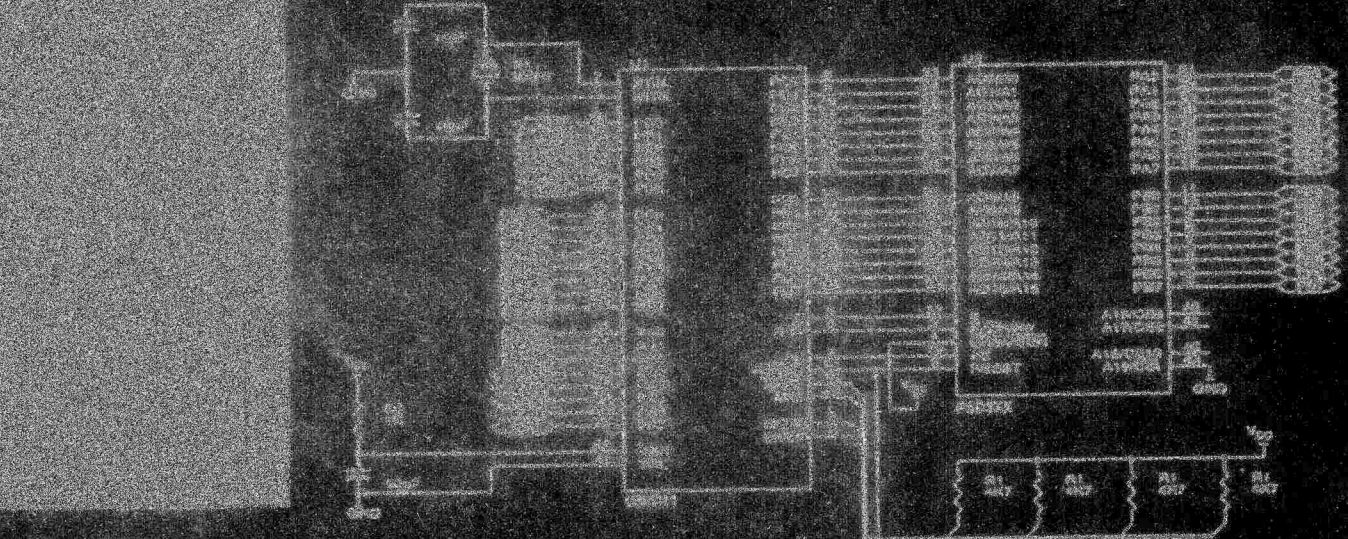


Figure 13

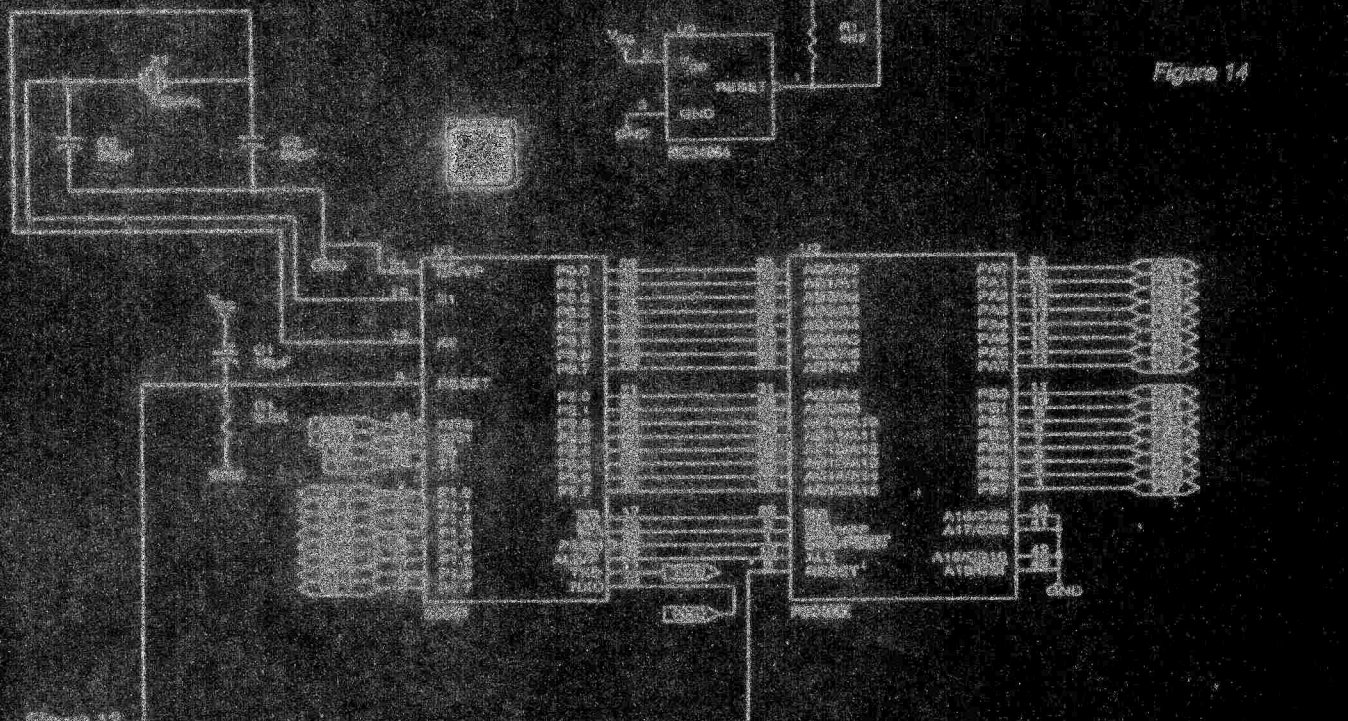


Figure 14

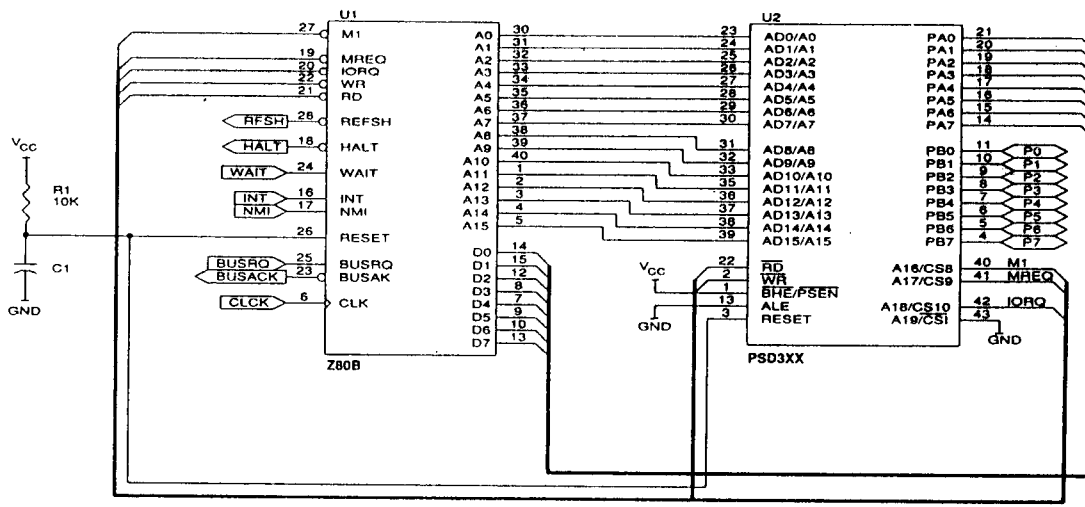


Figure 15

alors sur les bus d'adresses et de données, qu'il s'agisse d'un Z80 à 8 bits (figure 15), d'un 68000 à 16 bits (figure 16), ou de toute autre référence.

Le cas du 68000 est un peu moins favorable que les précédents, puisque tous les ports du PSD 3XX sont utilisés pour la liaison avec l'unité centrale, ce qui ne permet pas de créer de lignes d'entrée-sortie.

On peut résoudre le problème en associant deux PSD 3XX comme indiqué à la figure 17, artifice qui est d'ailleurs aussi applicable avec des microcontrôleurs. La figure 18 montre en effet que l'on peut ainsi augmenter massivement le nombre de ports d'entrée-sortie offerts par un 80C31, tout en doublant les ressources mémoire mises à sa disposition. Terminons avec le schéma de la figure 19, qui traite de l'association d'un PSD 3XX et d'un 6809, et surtout par le cas de la figure 20, détaillant l'adaptation du PSD 3XX sur un microprocesseur à bus multiplexé, en l'occurrence un 80186.

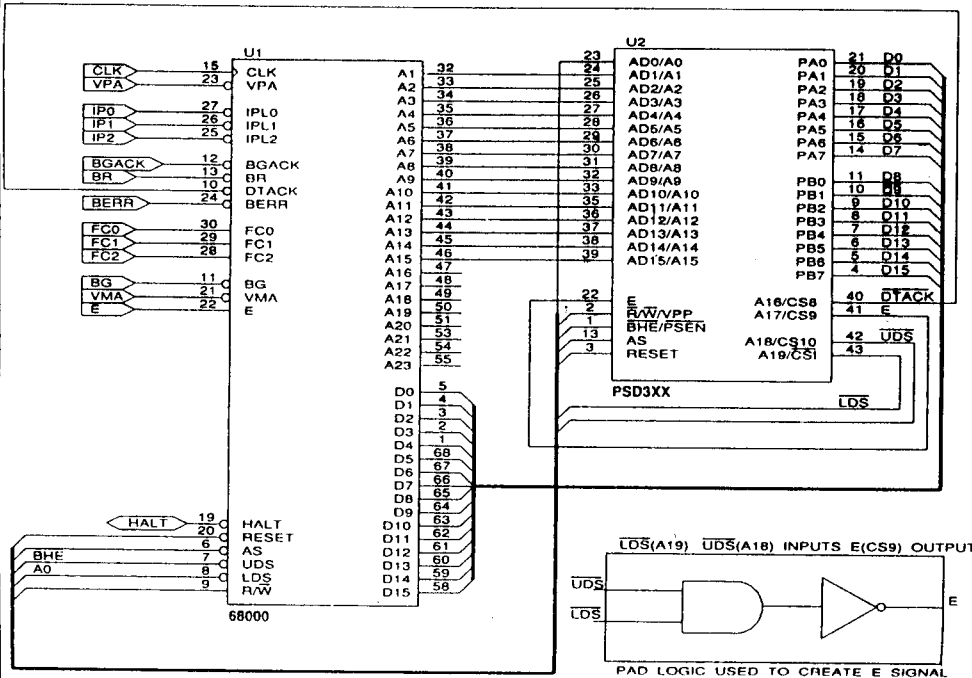


Figure 16

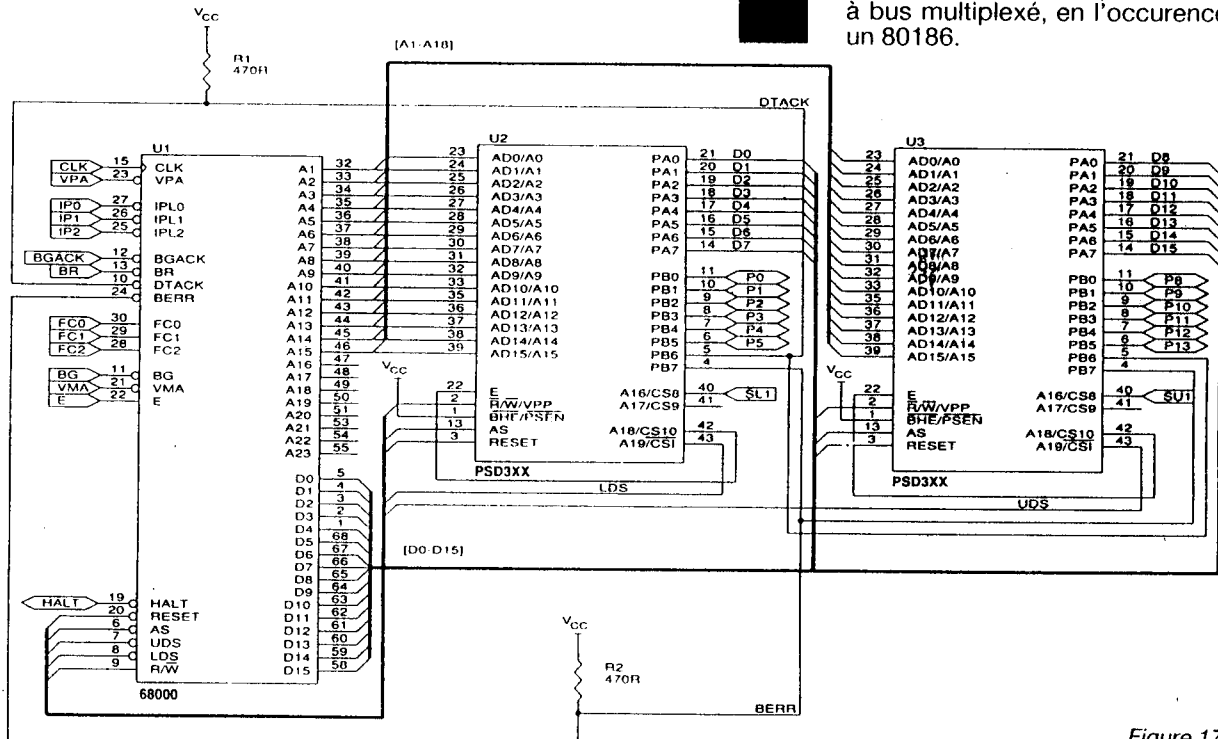


Figure 17

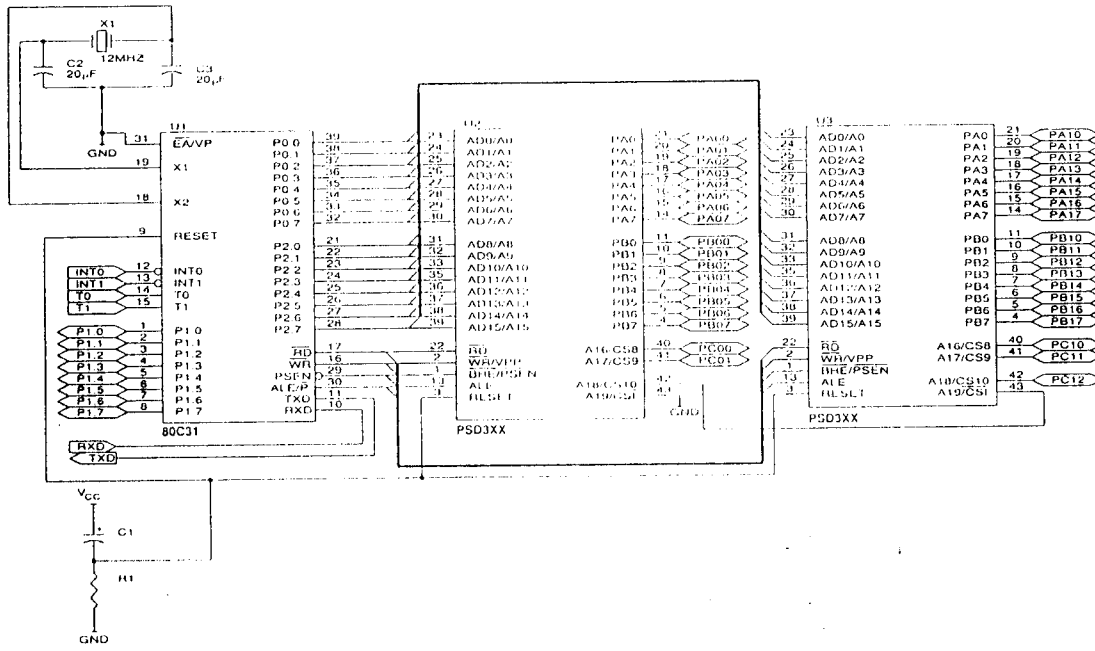


Figure 18

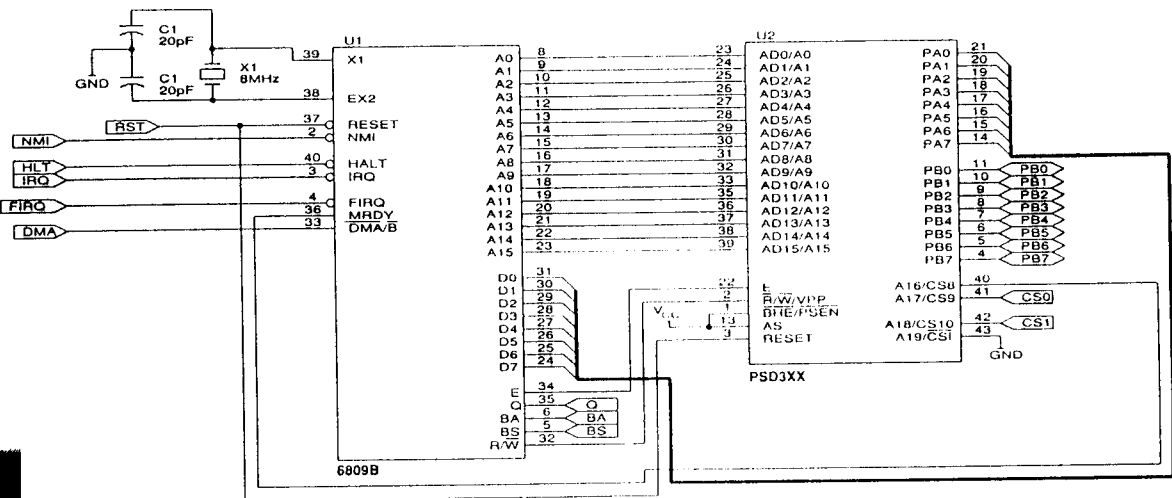


Figure 19

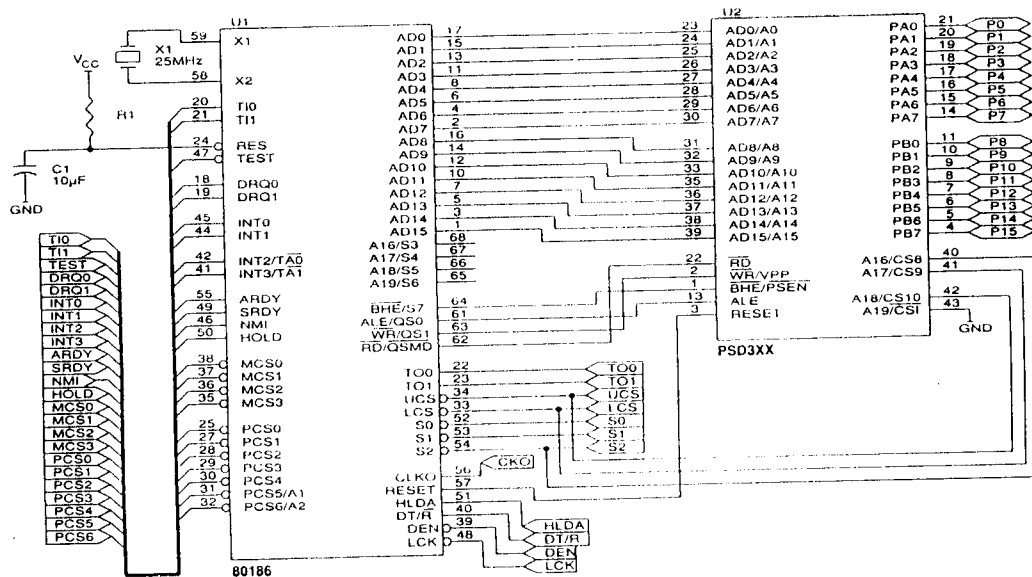


Figure 20

## LOGICIELS ET SYSTEMES DE DEVELOPPEMENT

Les PSD 3XX sont bien plus que des EPROM ou des PAL, et nécessitent pour leur mise en œuvre des outils de développement beaucoup plus sophistiqués qu'un éditeur de fichiers et un programmeur même dit "universel".

Votre premier logiciel de développement de PSD 3XX, "tournant" sur le premier PC venu, peut être obtenu sans frais car il fait partie du kit d'évaluation disponible sur demande qualifiée.

Sur une disquette 5" 1/4 haute densité ou trois disquettes de 360 K est ainsi offert un puissant logiciel permettant déjà de mettre intégralement sur pied des applications.

La procédure est particulièrement conviviale, puisqu'il s'agit essentiellement d'un échange de questions et de réponses en langage clair (anglais).

Il faudra cependant remplir quelques tableaux, à commencer par celui décrivant le plan mémoire que l'on souhaite instaurer. A ce stade, une parfaite connaissance du projet en cours et des compo-

sants utilisés est indispensable : pas question de répondre à peu près ou de "faire l'impasse" sur des demandes dont on n'est pas certain d'avoir bien compris le sens.

Tout comme l'écriture d'équations pour un compilateur de PLD, c'est véritablement là un travail de développeur averti et bien documenté.

Cette première étape produit un fichier disque qu'il faut encore compiler en un fichier "HEX" exploitable par le programmeur : soit un appareil de la marque, soit un modèle "agrée" (DATA I/O).

Le compilateur ne fait pas partie du logiciel d'évaluation gratuit, mais des deux "packages" disponibles moyennant finances : PSD-Silver (logiciel complet sans le programmeur), ou PSD-Gold (logiciel et programmeur "Magic-Pro").

Il est possible de développer avec le seul logiciel d'évaluation, puis de compiler et programmer ultérieurement en achetant la suite ou en s'adressant à un sous-traitant dûment équipé.

Cela mérite assurément d'essayer...

WSI est représenté en France par :

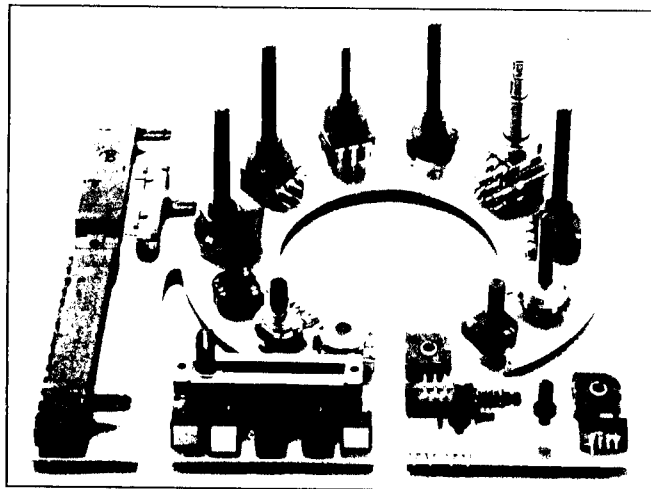
A2M  
B.P. 89  
78152 LE CHESNAY Cedex  
Tél. : (1) 39.54.91.13  
Fax : (1) 39.54.30.61

Patrick GUEULLE

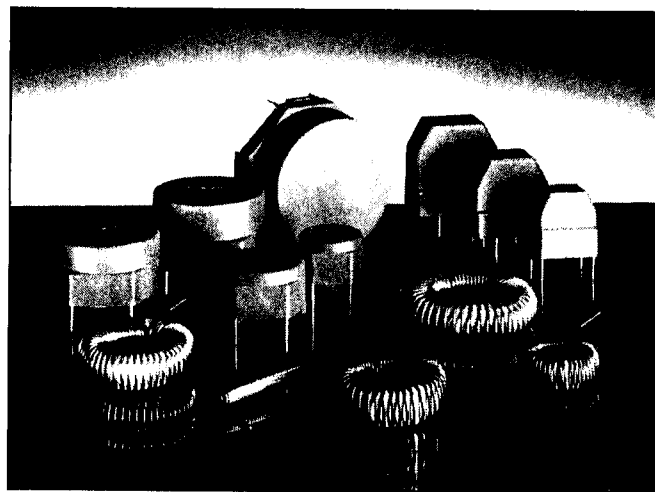


# Radiorhm

## POTENTIOMETRES & COMMUTATEURS



## INDUCTANCES ET FILTRES



**melek**

37, rue François-Arago - 93100 MONTREUIL - Tél. : (1) 48 58 94 09 - Fax : 48 58 70 04 - Télex : 233 414

Je désire recevoir gratuitement le nouveau catalogue.

POTENTIOMETRES & COMMUTATEURS  
 INDUCTANCES ET FILTRES

ERP 07/92

NOM ..... PRÉNOM .....

SOCIÉTÉ .....

ADRESSE .....

# PSD3XX : PROGRAMMATION ET APPLICATIONS

Lors des deux précédents Electronique

Radio-Plans, nous vous avons indiqué

comment fonctionnaient les PSD 3XX

et comment il fallait s'y prendre

pour transformer un schéma « classique »

en un schéma utilisant les PSD 3XX.

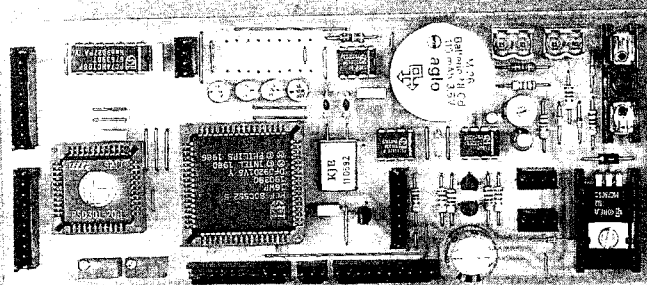
Notre but aujourd'hui consiste à vous

présenter les outils logiciels et de programmation permettant de

concrétiser votre concept et nous terminerons en vous donnant un

exemple de réalisation à usage général conçu autour du microcontrôleur

80C 552 bien connu.



Lors du dernier numéro, nous vous avons demandé de bien vérifier tous les «mappings» mémoire de vos chères réalisations de façon à être prêt dès le début de cet article. En voici les raisons profondes.

Une fois réalisé sur le papier le schéma électrique de votre réalisation, il ne reste plus qu'à le transformer «en hardware».

Or, pour effectuer ce passage, il est nécessaire d'utiliser un outil logiciel gourmand en informations provenant de votre fameux «mapping». Cet outil a pour finalité ultime, après compilation d'une foule de données de sélection d'options et de paramètres, de pouvoir fournir un «code» binaire digérable pour un programmeur ad-hoc permettant la programmation physique du composant choisi.

Ces outils fournis par la société créatrice de ces composants (WSI) ont pour nom MAPLE en ce qui concerne le logiciel et MAPPRO pour le programmeur dédié à ces composants. Il est à

professionnels classiques (DATA I/O, etc.) pourvu que le «release» adéquat existe !

De plus, une version de démonstration et d'évaluation comportant tous les fichiers (à l'exception du compilateur final, permet de se faire la main rapidement).

## LE LOGICIEL DE PROGRAMMATION

### Entrée en matière

Après avoir lancé le programme vous verrez apparaître un menu d'accueil vous proposant différentes options (**figure 1**).

Au lieu de vous présenter ces options de façon conventionnelle (rebarbative) du style F1 : bla bla bla , F2 : re bla bla, F3, etc... essayons d'être un peu plus intelligent (d'accord ça a des chances

dans lesquels on les utilise généralement.

Premièrement, on vérifie «bêtement» par F4 si le composant que vous désirez utiliser «PSD XYZ» est bien présent dans la «partlist» du logiciel... puis après être revenu au menu principal on passe au «partname» en déclarant l'élé de son cœur, par exemple le PSD 301.

Le fait d'appuyer sur la touche «enter» vous plonge dans l'angoisse du nouveau «menu» («MAIN MENU») dédié au PSD 301 choisi (**figure 2**).

Heureusement la **figure 3** retire beaucoup de sous-entendus en décrivant l'arborescence des différentes options car elle nous renvoie à des choses «hardware» connues. Déjà on se sent mieux.

Pour vous sentir encore plus chez vous, afin de personnaliser votre œuvre, vous avez tout intérêt à passer par la case «départ» dite de «project information» pour pouvoir l'identifier par la suite.



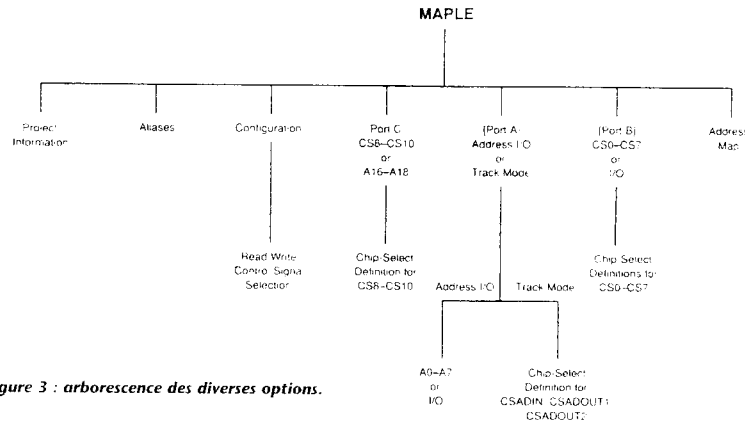


Figure 3 : arborescence des diverses options.

but de vous permettre de baptiser de noms qui vous seront plus sympathiques tous les signaux de noms barbares présents jusqu'alors sur les broches (tels que A19-A16 et les CS0-CS7).

Un exemple vous est présenté figure 4.

Il est à noter que par la suite lors des configurations des ports A, B, C, etc., vous retrouverez vos noms propres. Tous ces hors-d'œuvres étant effectués, après être remontés au «main menu» on peut maintenant passer au vif du sujet en commençant par le menu de «configuration».

## Configuration du type de microcontrôleur

Bien que cela ne soit pas dit explicitement, le menu de configuration permet en fait de sélectionner le type de microcontrôleur et le type d'architecture que vous désirez utiliser.

La figure 5 donne l'exemple d'une configuration pour un microcontrôleur de la famille 8xC 51 (pour l'application présentée 8xC 552) avec un bus adresses-données multiplexées de 8 bits, reset à l'état haut, ALE, etc.

Deux petites remarques : Etant donné que les circuits de la famille PSD 3xx peuvent supporter de très nombreux microcontrôleurs de toutes marques et de structures de bus très variées, les menus sont auto-évolutifs selon que vous choisissez des bus de largeur 8 ou 16 bits et vous proposent des options adaptées notamment en ce qui concerne les possibilités qu'offrent les READ, WRITE et PSEN imbriqués. A vous de choisir vos options mais... c'est vous qui voyez !

## Configuration des ports

### Le port C

Vous pensiez commencer par le port A, eh bien non ! On vous propose d'abord le port C.

La figure 6 vous présente ce menu très restreint car il n'y a que trois bits à définir sur ce port et votre choix ne peut s'effectuer que sur un choix limité de signaux (A16-A18 et/ou CS8-CS10), par exemple le PC1 en CS9.

répondre le Chip Select (CS) en question en passant au menu suivant via F3.

Chaque rangée de paramètres présente correspond à une équation logique écrite sous forme d'un terme «produit».

Un exemple vous est donné figure 7 dans lequel les cases ont été remplies avec des «1», «0» et «x» où :

«1» signifie que le signal considéré sera actif à l'état haut

«0» signifie que le signal considéré sera actif à l'état bas

«x» ou «vide» signifie inopérant.

et dans notre exemple :

$$PC1 = CS9 = A19 \cdot A18 \cdot A16 \cdot A15 \cdot A14 \cdot A12 \cdot A11 \cdot E \cdot R/W$$

### Le port A

Après retour à la case départ «main menu» vous courez au menu du port A.

Deux options vous sont proposées. La première consiste à configurer les «entrées/sorties» de ce port et l'autre à le configurer en «suiveur» pour des extensions externes du port. Ce mode de fonctionnement ne faisant pas partie de nos préoccupations actuelles, nous ignorons pour l'instant cette option (nous renvoyons momentanément les lecteurs désirant utiliser ce mode à la spécification du concepteur du logiciel).

Alors en route pour le «address 1/0» menu

La figure 8 vous le présente et il est suffisamment explicite en lui-même pour éviter tout commentaire concernant les sorties CMOS ou Open Drain.

### Le port B

Le premier choix est dédié à la configuration hardware du port, figure 9 et le second, après avoir défini un PBx en CSx en appuyant sur F3, figure 10 vous amène à définir à nouveau votre équation logique de la même façon que précédemment.

## Le Mapping des adresses

Le menu «address map» du «main menu» va permettre de configurer le (votre) mapping des EPROMs, SRAM et CSIOPORT (voir figure 11).

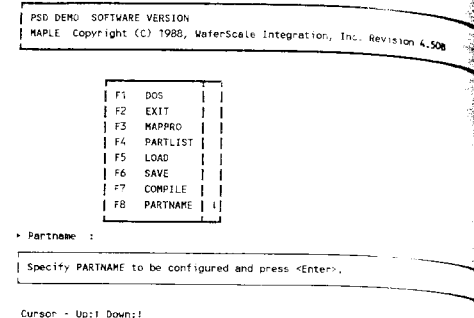


Figure 1

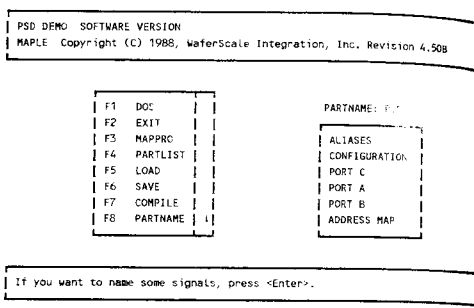


Figure 2

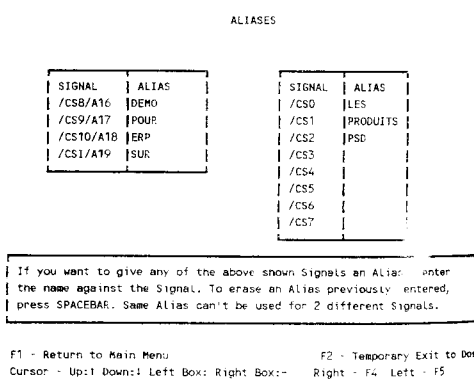


Figure 4

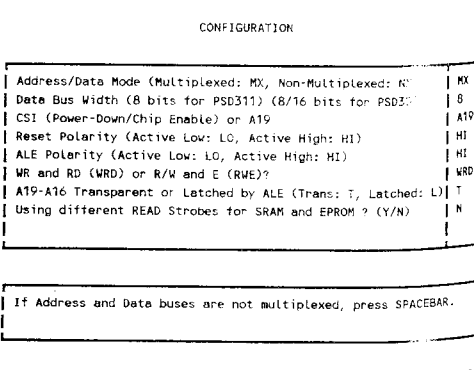


Figure 5

Nous ne serons hélas pas à vos côtés, lorsque vous remplirez vos propres cases mais nous pouvons vous donner quelques explications pour sa mise en œuvre.

Bien évidemment, si vous êtes arrivé à ce stade des tableaux de programmation, c'est que votre étude n'attend plus que de vous être présentée et



PORT C

PIN	CS/A1
PCD	A16
PCT	CS9
PC2	A18

Configure all the 3 pins before going to any CS Definition.

If you want to configure PCD as Chip Select, press SPACEBAR.

1 - Return to Main Menu  
3 - Goto CS Definition

F2 - Temporary exit to Dos  
Cursor - Up: Down: Left: Right: |

Figure 6

PORT C

PIN	CS/A1
PCD	A16
PCT	CS9
PC2	A18

CHIP SELECT DEFINITION POUR											
A19	A18	A16	A15	A14	A13	A12	A11	RD	WR		
0	1	1	1	1	1	0	0	X	1	1	0

CS definition is the NOT of the product term (row). Enter 1 to select Active High signal, 0 to select Active Low signal, X to mean 'don't care', SPACEBAR to erase. Enter values in columns relevant to your application; other blank columns will be treated as 'don't care's.

1 - Return to PORT C  
Cursor - Left: Right: |

Figure 7

PORT A

Prev Config:- ADDRESS/IO TRACK MODE

If you want to configure PORT A pins individually as Address or I/O bits, press <Enter>.

1 - Return to Main Menu  
3 - Goto CS Definition

F2 - Temporary exit to Dos.  
Cursor - Up: Down: Left: Right: |

Figure 8

PORT B

PIN	CS/I/O	CMOS/OD
PB0	CS0	CMOS
PB1	CS1	CMOS
PB2	CS2	CMOS
PB3	CS3	CMOS
PB4	CS4	CMOS
PB5	CS5	CMOS
PB6	CS6	CMOS
PB7	CS7	CMOS

SIGNAL	ALIAS
/CS0	LES
/CS1	PRODUITS
/CS2	PSD

If you want to configure PB0 as I/O press SPACEBAR.

1 - Return to Main Menu  
3 - Goto CS Definition

F2 - Temporary exit to Dos  
Cursor - Up: Down: Left: Right: |

Figure 9

compilation, assemblage et linkage (les deux mamelles de la programmation micro...) et création de fichiers «objet» en Intelhex pour faire joli.

Bref tout cela existe et même, pour être très professionnel, vous vous êtes créé un «directory» spécial du type PCD dans lequel vous avez disposé de

pour la suite des événements. Si cela n'était pas le cas une petite descente sur le DOS via EXIT s'impose d'urgence pour mettre tout cela au propre et personne ne le saura. Donc, nous pouvons remplir les cases.

## Pour les mémoires EPROM

Pour positionner le début de votre code (les ESx) vous pouvez le faire par bloc soit en binaire soit en hexa (mais pas les deux en même temps sur un même bloc) et indiquer pour ce bloc le nom du fichier «objet» que le compilateur du programme MAPLE ira chercher dans le même directory lors de son travail final de «moulinage» pour créer le fichier assimilable et compréhensible par le programmeur du composant.

Dans l'exemple présenté, il y a deux fois le même nom sur deux lignes. Ne vous inquiétez pas, il ne reprendra pas deux fois le début du même code à des adresses différentes, il continuera de vider la suite de votre fichier jusqu'à épuisement.

La remarque du dessus peut vous paraître puérile, mais, sachant que certains types de cette famille peuvent posséder jusqu'à 128 Koctets de code (PSD303 et PSD313), comment la «pagination» du PSD xxx est-elle gérée au-delà des 64 Ko fatidiques des 8xC XXX ? Voici une bonne question !

Dans ces versions, le circuit contient un «Page Register» qui a pour mission d'assurer la fonction de pagination automatiquement lorsque votre code sera pris en compte lors de la compilation finale et ce sans que vous ayez à vous préoccuper de réaliser des acrobaties matérielles et/ou logicielles.

## Pour la RAM et la CSIOPORT

Rien de particulier. Vous voici arrivés au bout de vos peines et aussi avant de tout détruire rien ne vaut une bonne sauvegarde via SAVE qui produit deux fichiers : un .SAV de sauvegarde un .SV1 de reporting. Le .SAV sera rechargeable via le LOAD pour continuer, améliorer, modifier le projet et le .SV1 pourra être édité sous DOS via PRINT pour retrouver en clair les «aliases», les «équations logiques»,...

## La compilation et programmation du composant

Une passe de compilation, quelques insultes (messages d'erreurs) du fait de recouvrement d'adresses involontaires, quelques corrections et tout sera terminé après passage sur le programmeur du labo. Il ne reste alors qu'à tester la carte. Quelle carte ?

## CARTE APPLICATIVE ET D'EVALUATION

PORT B

PIN	CS/I/O	CMOS/OD
PB0	CS0	CMOS
PB1	CS1	CMOS
PB2	CS2	CMOS
PB3	CS3	CMOS
PB4	CS4	CMOS
PB5	CS5	CMOS
PB6	CS6	CMOS
PB7	CS7	CMOS

CHIP SELECT DEFINITION LES											
A19	A18	A16	A15	A14	A13	A12	A11	RD	WR		

CS definition is the NOR of the product terms (rows). Enter 1 to select Active High signal, 0 to select Active Low signal, X to mean 'don't care', SPACEBAR to erase. Enter values in columns relevant to your application; other blank columns will be treated as 'don't care's.

F1 - Return to PORT B  
Cursor - Up: Down: Left: Right: |

Figure 10

ADDRESS MAP

	A19	A18	A17	A16	A15	A14	A13	A12	A11	SEGMENT	SEGMENT	FILE	FILE	FILE NAME		
	START	STOP	START	STOP	START	STOP	START	STOP	START	STOP	START	STOP	START	STOP		
ES01	0	0	0	0	0	0	0	0	0	0	0	10000	1FFF	0	0	DEM1_ERP.HEX
ES11	0	0	0	0	0	0	0	0	0	0	0	10000	1FFF	1000	0	DEM1_ERP.HEX
ES21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ES31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ES41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ES51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ES61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ES71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
RS01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
CSPT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

ALIANCES: A19=SR A18=ERP A17=POUR A16=DEMO  
Fill in A19-A11 (Binary) or SEGMENT START (Hex); and FILE(START, STOP) and FILE NAME. Use SPACEBAR to erase any field value.  
F1 - Return to Main Menu F2 - Temporary exit to DOS F3 - Goto Help  
Cursor - Up: Down: Left Col: Right Col: | Right - F4 Left - F5

Figure 11

PSD 3xx, Jean-Pierre BILLIARD et moi-même avons décidé de vous offrir une réalisation passe-partout capable d'être utilisée soit directement pour de nombreuses applications standards soit servir de carte d'évaluation de ces produits.

Les schémas électrique et d'implantation de cet ensemble sont donnés figures 12 et 13 et n'appellent pas de commentaires particuliers autres que ceux déjà fournis lors de la description ERP numéro..., car comme vous l'avez certainement remarqué nous nous sommes à nouveau servis du même microcontrôleur (c'est tellement plus simple de travailler bien et une seule fois !!!).

A noter que c'est par pure protection que nous avons laissé un boîtier de portes NAND qui aurait dû (et pu) être englouti par le PSD 3xx.

Bref, cette fois encore nous avons gagné en nombre de composants, en surface de cuivre, en EMC, en temps de développement, en souplesse de conception, en standardisation de nomenclature, en temps passé en inventaire, etc. et donc en coût. (voir photo comparative entre l'ancienne et la nouvelle figure 14 surtout au niveau de l'environnement direct de la CPU).

Par les temps qui courent, ce sont des choses remarquables n'est-ce pas ? Voici terminée la prise de contact avec ces circuits périphériques programmables pour microcontrôleur. A bientôt.

Dominique PARET  
Avec l'aimable



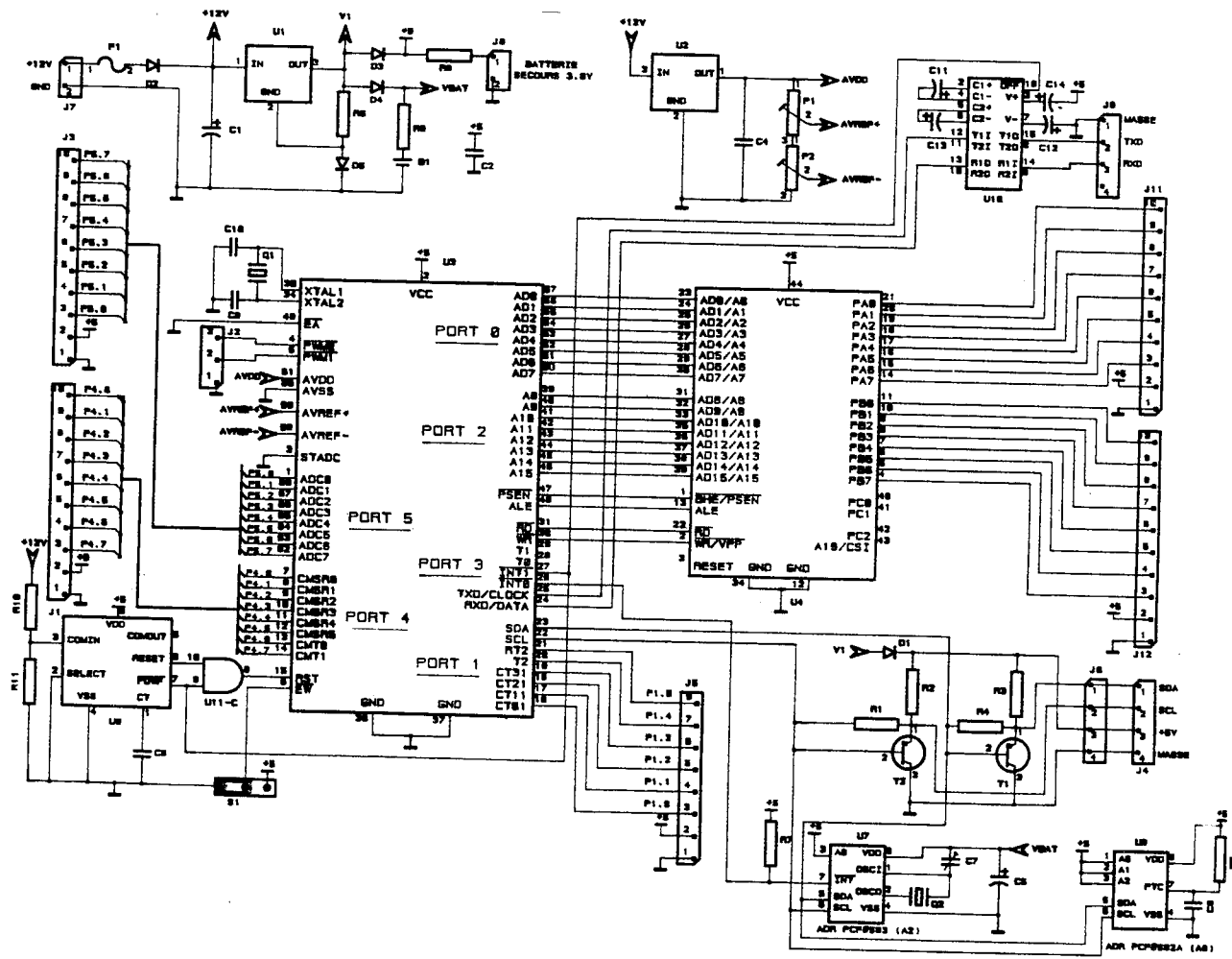
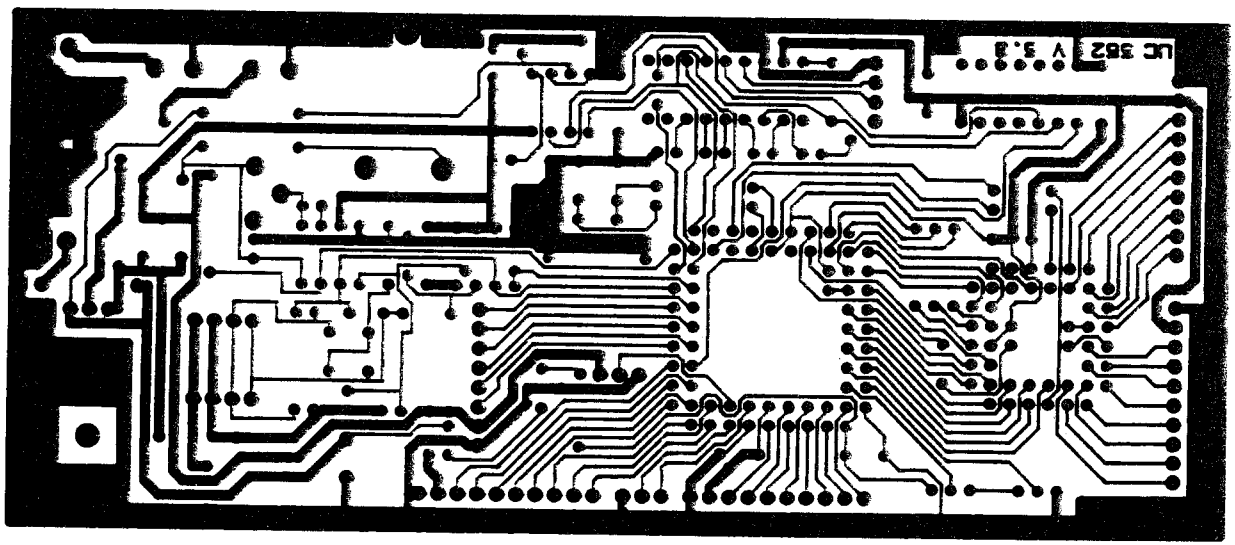


Figure 12 : le nouveau schéma de la carte 80C552 avec le PSD301.



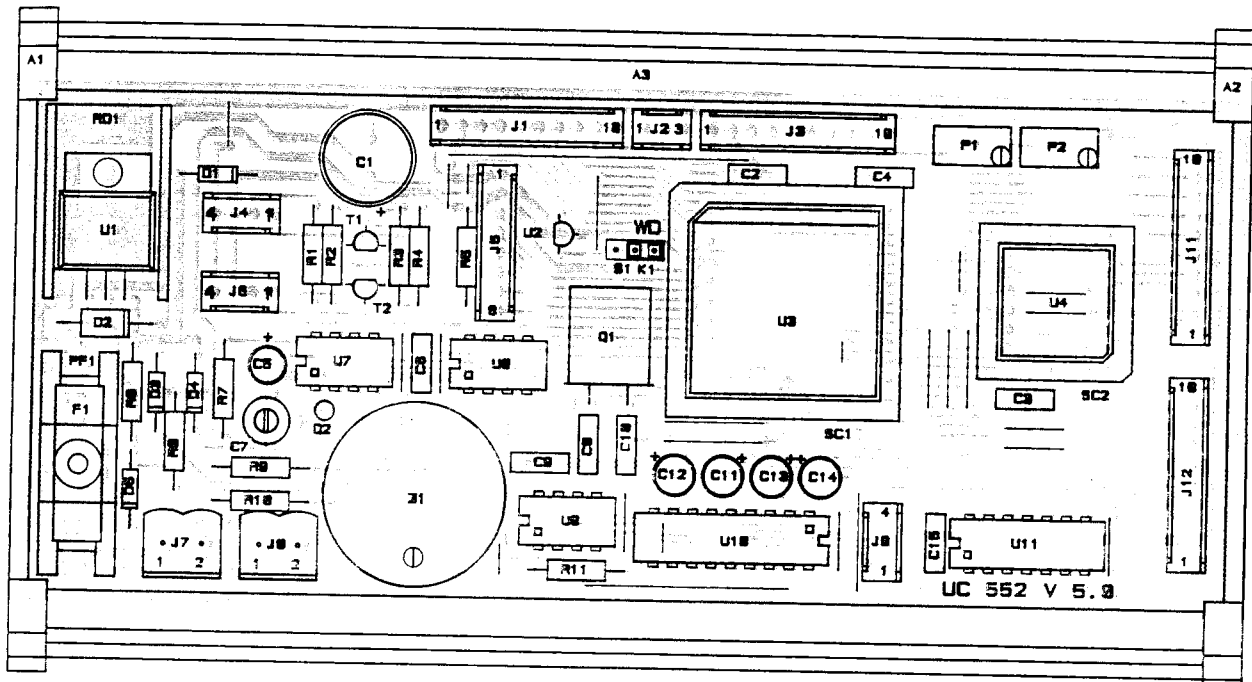


Figure 13 b : l'implantation de la nouvelle carte mettant en œuvre le PSD301.

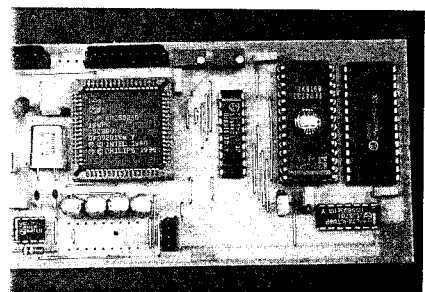
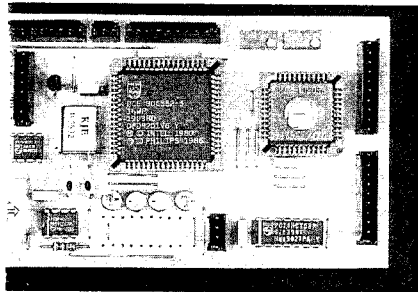


Figure 14 : l'ancienne et la nouvelle versions côte à côte.

## SIMULATION ÉLECTRONIQUE MIXTE

### IsSpice3 affiche les signaux pendant les calculs !

- Entrée de schémas
- Simulation SPICE
- Bibliothèques de modèles
- Monte Carlo
- Oscilloscope logiciel
- Prix ABORDABLES

Simulation par IsSpice3 d'un convertisseur A/D

**ENVIRONNEMENT INTÉGRÉ - IMPLANTATION ASSISTÉE**

Le langage de description de circuits imprimés est le langage de description de circuits imprimés.

**SPICENET**, outil dédié à la saisie de schémas.

**RESPIICE**, qui comprend, entre autres, des bibliothèques de composants très complètes.

**ISSPICE**, noyau de calcul SPICE, effectue des analyses AC, DC, transitoire, time, noise, pole-zero,.... La nouvelle version IsSpice3, basée sur le SPICE 3E.2 de Berkeley, permet en plus d'évaluer des expressions booléennes et des multipôles définis par une simple équation.

**INTUSCOPE**, oscilloscope logiciel, affiche, manipule et effectue des opérations sur les courbes obtenues aux différents nœuds après simulation.

Il existe plusieurs versions ICAPS, à partir de 9800 F H.T.

ICAP est un produit de la société INTUSOFT.

**EXCEM**  
Département  
Produits Informatiques  
12, chemin des Hauts de  
Clamart

# CAO

sur PC/AT et compatibles

..PRIX...PRIX...PRIX...PRIX.

"Boardmaker 1" 834,74 F ht  
"Boardmaker 2" 2990 F ht  
"BoardRouter" 2990 F ht

"BoardMaker 2 + BoardRouter"  
EN PROMOTION 4950 F HT  
(offre limitée)

## COMMENT ECONOMISER ?

### SANS COMPROMIS !

# "BOARDMAKER"

logiciels de saisie de schéma et dessin de circuits imprimés sur PC XT/AT

Produit des laboratoires de recherche de Cambridge, le Silicon Valley anglaise, Boardmaker est partiellement adapté aux applications professionnelles les plus pointues: cartes multicouches, composants CMS, génération de fichiers pour production sur machines à commande numérique, Ecrans CGA, EGA, VGA, avec "ZOOM" à 7 grossissements différents permettant d'afficher toute la carte de 43cmx43cm maxi, ou une zone de sept carreaux de la grille de 2,54mm ! Sorties sur Imprimantes matricielles, lasers, tables-traçantes, Gerber et NC drill. Deux logiciels pour le prix d'un seul car Boardmaker permet de réaliser la saisie de schéma et le dessin de circuit imprimé. Bibliothèques de symboles graphiques extensibles par l'utilisateur. Souris et DD recommandés. 640 K min.

**Boardmaker.. une famille évolutive!**

Commencez par Boardmaker 1 et évoluez plus tard vers Boardmaker 2 et même BoardRouter...vos fichiers sont transmissibles de l'un à l'autre...pas de travail perdu ! Livrable sur stock.

**Boardmaker 1: 990.00 F TTC franco**

NOTE: Boardmaker 1 ne traite pas les netlists mais contient tous les drivers professionnels Gerber, drill etc Voir le Compte rendu d'essai Radio-Plans Octobre 1990.

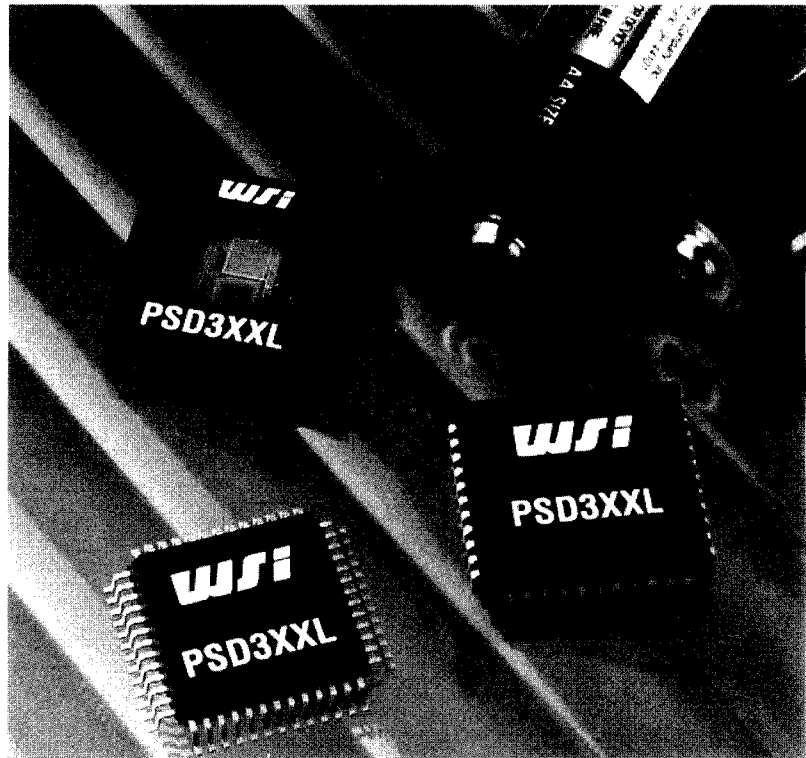
**BON DE COMMANDE. DE LA DISQUETTE DEMO AVEC MANUEL EN FRANCAIS.**  
Ci- joint mon chèque de 50 Frs TTC franco (même prix pour BM1, BM2 ou BM2 + BR)  
NOM: \_\_\_\_\_ ADRESSE: \_\_\_\_\_  
Format désiré: 3,5"5,25"

# Les périphériques programmables PSD 3XX WSI

Lors des derniers articles, nous avons traité des problèmes levés par la CEM (ERP 547), en orientant nos propos principalement autour des composants tels que les microcontrôleurs et des circuits imprimés immédiatement environnants (routages, connecteurs, ...). Nous avons insisté plus particulièrement sur le fait que les «dimensions» mises en œuvre devaient être les plus «courtes» possible.

En plus d'exigences CEM, il existe à ce jour de nombreuses autres raisons (réduction de surface, de volume, de masse, de souplesse d'application, d'opportunité, de facilité de conception, de coût,...) qui font converger les fabricants de composants vers l'introduction sur le marché de certaines nouvelles familles de circuits intégrés.

Parmi ceux-ci, la société américaine WSI a développé une gamme de produits dits «Programmable Peripherals» pour applications à microcontrôleurs et satisfaisant tous ces souhaits. De plus, il y a quelques mois, le PHILIPS semiconductors a annoncé qu'il assurait mondialement la double source de cette famille de produits baptisés PSD3XX.



## PÉRIPHÉRIQUES PROGRAMMABLES POUR MICROCONTRÔLEURS

### But de l'opération

Le but final de cette famille de composants consiste à satisfaire les principaux points suivants :

a) réduire au strict minimum le nombre de composants (toute la «glue» logique) autour du microcontrôleur à l'aide d'un «mono-chip» et donc réduire les coûts, volumes, EMC,...

b) du fait que ces circuits sont programmables, ceci doit permettre (à même implantation physique sur le circuit imprimé) de reconfigurer facilement et rapidement les fonctionnalités «hardware» d'une carte en la rendant de ce fait plus «universelle», donc en offrant des solutions pour satisfaire rapidement de nouvelles opportunités commerciales (et ne plus entendre crier et pleurer les équipes de marketing pour causes de «pertes» d'affaires !).

Notre plan de présentation de ces produits s'effectuera en trois grandes étapes.

Ce numéro vous présentera ces produits et leurs possibilités générales d'application.

Le numéro suivant vous apprendra, à l'aide d'un cas précis d'application complexe déjà existant, comment réduire à la «cuisson» la structure de son schéma au strict minimum, et nous vous offrirons un «cuivre» d'application à usage général.

Enfin, pour terminer cette «série», nous donnerons la marche à suivre pour programmer ces circuits.

### L'architecture des PSD3XX

L'architecture générale de l'un des composants de la famille des PSD3XXX est donnée aux figures 1 et 2 et nous allons maintenant assurer la visite guidée.

Le schéma bloc de la figure 1 donne l'architecture globale d'un tel circuit. Celui-ci comprend la quasi-totalité des périphériques classiques nécessaires à l'élabo-

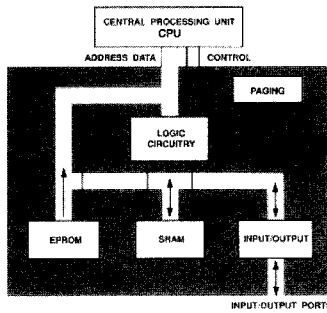


Figure 1

ration d'une application ayant une CPU à microcontrôleur, c'est-à-dire :

- a) des mémoires, de la RAM d'une part et de la mémoire programme (EPROM UV ou OTP) d'autre part, y compris leurs possibilités de pagination ;
- b) toute la logique nécessaire au traitement des signaux classiques de «service» (Read, Write, Adress Latch Enable, Chip Select,...).
- c) de la logique programmable (de type PLD) pour satisfaire la majorité de vos fantaisies (souvent nécessaires) autour du microcontrôleur, citons par exemple «codage» ou «décodage» de nombreuses interruptions ;
- d) et enfin pour terminer ces généralités, la reconstruction de ports d'I/O pour ne pas pénaliser les ressources propres du microcontrôleur et pouvoir assurer des «emboitements» de nombreux circuits entre eux tant «horizontaux» (en largeur de bus) que «verticaux» (en profondeur de sous-systèmes associés).

Reprenons dans le détail tout ceci à l'aide de la **figure 2**. Premièrement et d'abord cette famille de circuits possède 44 broches (PLCC QFP...), voir **figures 3 et 4**. Déjà ce nombre définit l'arsenal des broches d'entrées/sorties disponibles pour nos fanstasmes futurs !

### Le bus d'adresses/données

16 broches sont réservées à cet usage et c'est déjà ici que surgissent les fausses difficultés.

Sont-ce 2 x 8 bits ? Sont-ce 16 bits d'adresses ou de données ou d'adresses/données multiplexées ou etc. ?

Cette famille de circuits a été élaborée pour satisfaire tant les microcontrôleurs 8 bits à bus «adresses-données» séparées ou encore les bus «adresses/données» multiplexés et les microcontrôleurs 16, 32, 64 bits en les cascadiant.

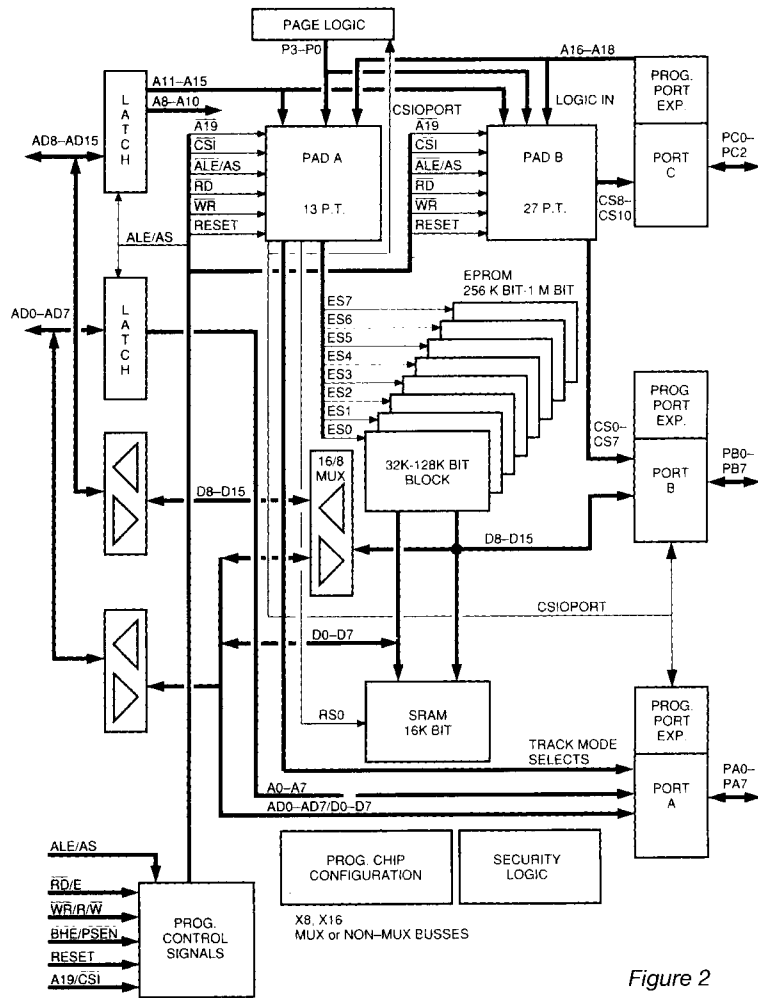


Figure 2

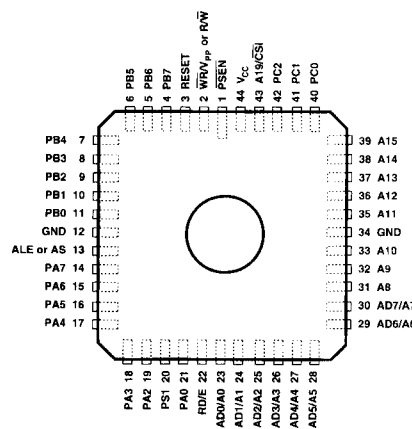


Figure 3

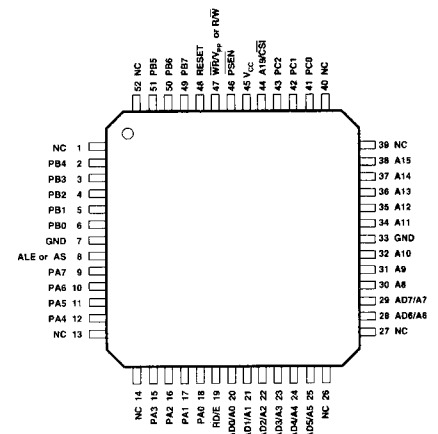
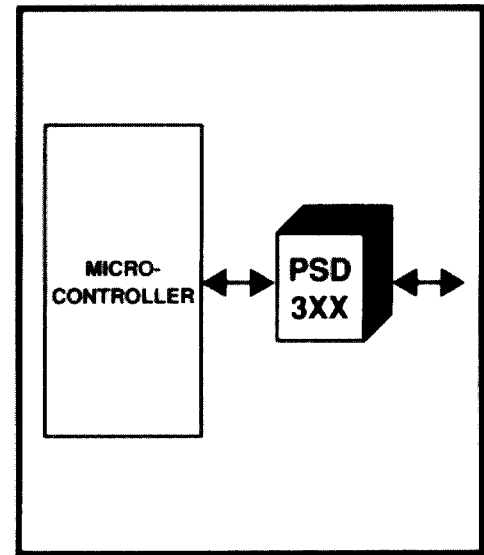
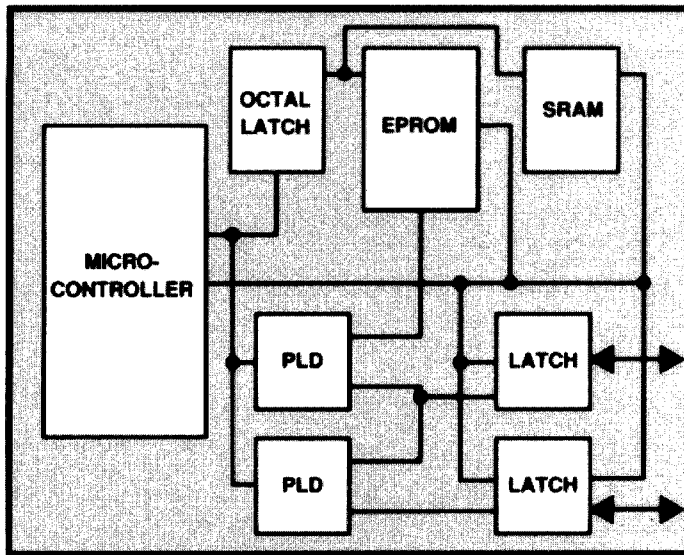


Figure 4

Si l'on regarde attentivement la figure on s'aperçoit que le circuit contient les «latches» d'adresses dont auront besoin tous les microcontrôleurs ayant des bus multiplexés temporellement, par exemple ceux de la famille 80C51.

Toute liberté vous sera donc offerte au niveau du choix du type des microcontrôleurs. Afin de ne pas tomber dans l'encyclopédie

en 18 volumes décrivant l'interfaçage à tous les types de microcontrôleurs, nous nous restreindrons dans nos explications aux applications de la famille 80C51 car abondamment décrite dans la revue et nous vous donnerons quand cela sera nécessaire les grandes lignes pour des applications de type 68HC... ou encore en 16 bits les 68 000 ou 90C. Evidemment, il faut bien que le



circuit intégré comprend avec quel microcontrôleur il travaille et comment il doit s'organiser pour traiter les signaux de service.

A cet effet, un certain nombre de broches ayant fréquemment de doubles fonctionnalités ont été prévues de façon à pouvoir, lors de la programmation, personnaliser le type de CPU avec qui il est censé fonctionner.

Pour vous faciliter la tâche, le logiciel de programmation du circuit, comme nous le verrons plus tard, s'occupe pratiquement de tout si vous lui déclarez la famille de micro dont il s'agit.

Par exemple pour le 80C 51, voir la figure 5, pour les 68 000/90C xxx, la figure 6.

De plus, en cas de «bêtes» étranges, la polarité de certains signaux peut être modifiée. Bref, après tout ce petit ménage de démarrage pas trop compliqué à effectuer, nous allons passer aux choses plus nobles.

Configuration	Bits	Function
CDATA	0	8-bit data bus
CADDRDAT	1	Multiplexed address/data
CRRWR	0	Set RD and WR mode
CA19/CS1	0	Set CS1 input power-down mode
CALE	0	Active HIGH ALE
CRESET	1	Active HIGH RESET
COMB/SEP	1	Code and data memory separate
CPAF2	0	Input/Output Port A
CPAF1	00H	Input/Output Port A (0-7)
CPBF	FFH	Input/Output Port B
CPCF	111B	Outputs CS8-CS10
CPACOD	00H	Configure CMOS outputs Port A
CPBCOD	00H	Configure CMOS outputs Port B
CADDHLT	X	"Don't care" for latched A16-A19
CSECURITY	0	No security

Figure 5

Configuration	Bits	Function
CDATA	1	16-bit data bus
CADDRDAT	0	Non-multiplexed address/data
CRRWR	1	Set R/W and E control inputs
CA19/CS1	1	Enable A19 input
CALE	X	ALE polarity set at "don't care"
CRESET	0	Active LOW RESET
COMB/SEP	0	Combined memory mode
CPAF2	X	"Don't care" Port A
CPAF1	XX	"Don't care" Port A
CPBF	X	"Don't care" Port B
CPCF	110B	Enable A16 and A17 Out. A18 In
CPACOD	00H	Configure CMOS buffers Port A
CPBCOD	00H	Configure CMOS buffers Port B
CADDHLT	0	Transparent A16-A19
CSECURITY	0	Security off

Figure 6

### Les accès mémoire

Une grande partie de ce travail est effectuée à l'aide du «PAD A» (PAD pour Programmable Array Decoder) dans lequel, comme vous pouvez le remarquer sur la figure 2, tous les signaux de service sont insérés.

Prenons le cas d'application de ces circuits avec la famille 8x C xxx. Tout d'abord, les figures 7 et 8 indiquent comment on peut réaliser les deux cas d'application classiques qui sont :

a) le cas où les deux espaces mémoire RAM et (EP)ROM sont des

espaces totalement séparés (c'est-à-dire pas de code exécutable dans la RAM) ;

b) le cas où ces espaces sont «combinés», c'est-à-dire lorsque l'on pourrait ou pas exécuter du code si celui-ci était disposé dans la RAM. A cette occasion, on peut remarquer que ce cas devient de plus en plus fréquent car nombre d'applications commencent à utiliser des mémoires Flash E(2) PROM en lieu et places des RAM et E(2)PROM pour réaliser des systèmes «téléchargeables».

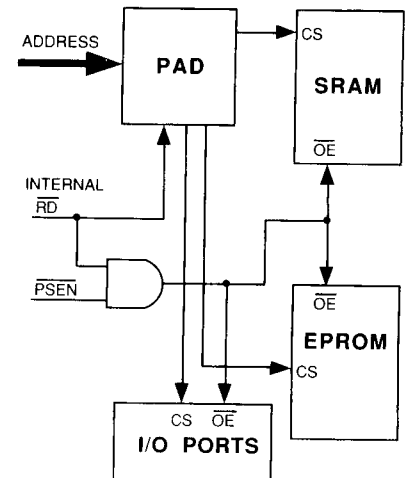


Figure 7

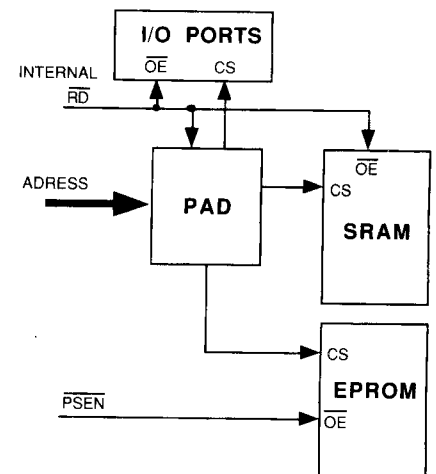
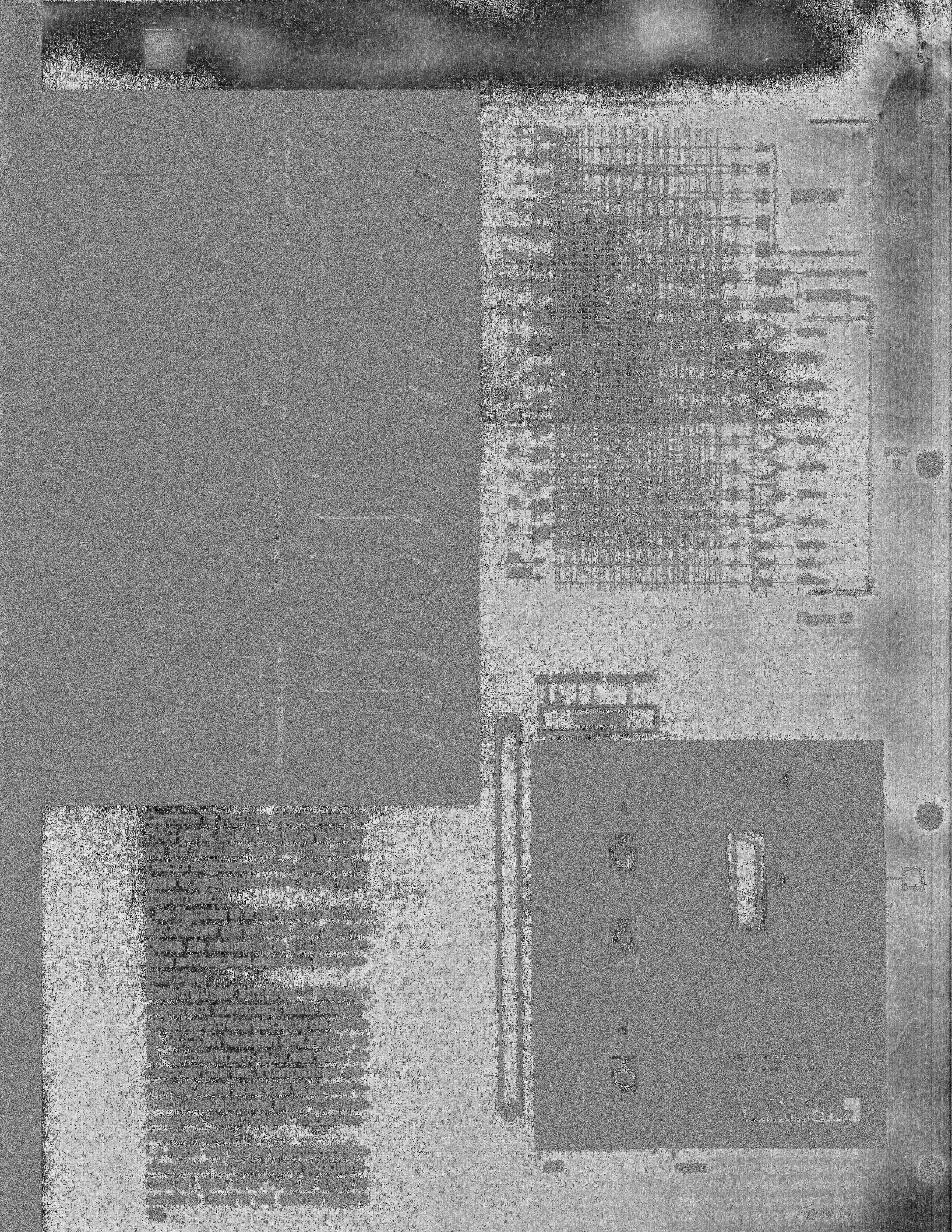


Figure 8





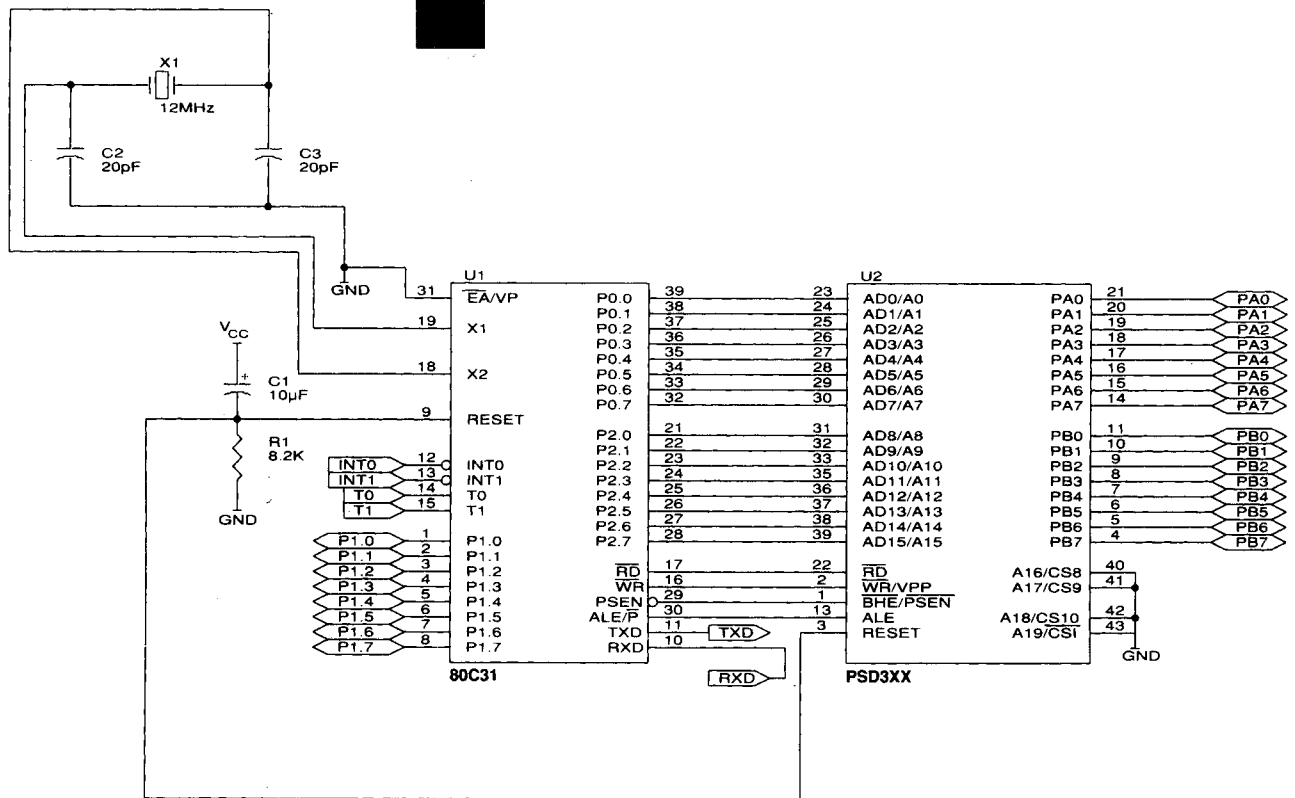


Figure 17

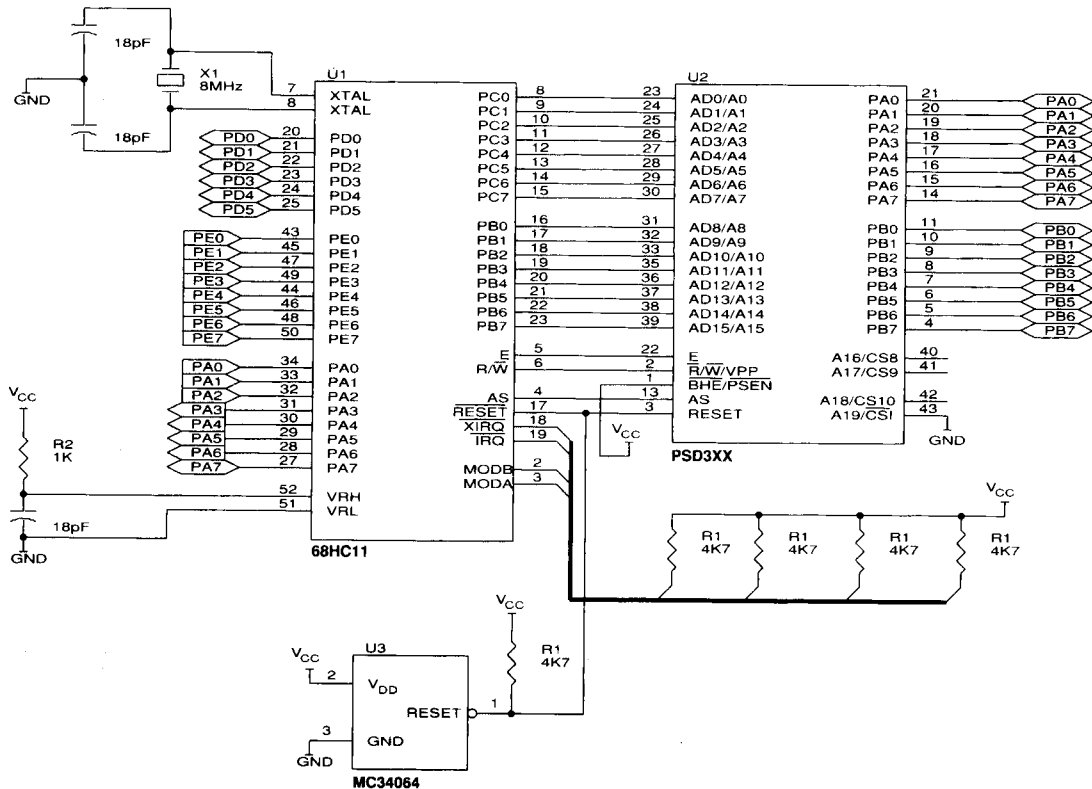


Figure 18

### Pagination des plans mémoire

Un exemple de pagination est donné figure 16 où il est supposé que l'utilisateur a besoin des 128 K octets de «ROM» et 32 K de RAM, et dans lequel il a été supposé la présence de trois périphériques «mappés» dans le plan mémoire.

La figure vous indique sommairement comment les broches de sorties liées aux différents PAD ont été attribuées.

Lors de l'exemple précis que nous avons choisi de vous proposer dans le prochain numéro, nous vous indiquerons par le détail comment réaliser tout ceci. Chaque chose en son temps.

### Schémas d'applications de base

Une fois tout cela effectué, les schémas de base se trouvent souvent réduits à leur plus simple expression. Ça devient très triste, l'électronique...

Trois exemples concernant les familles génériques 8x C xxx, 68 HC xxx et 6800/90C xxx sont

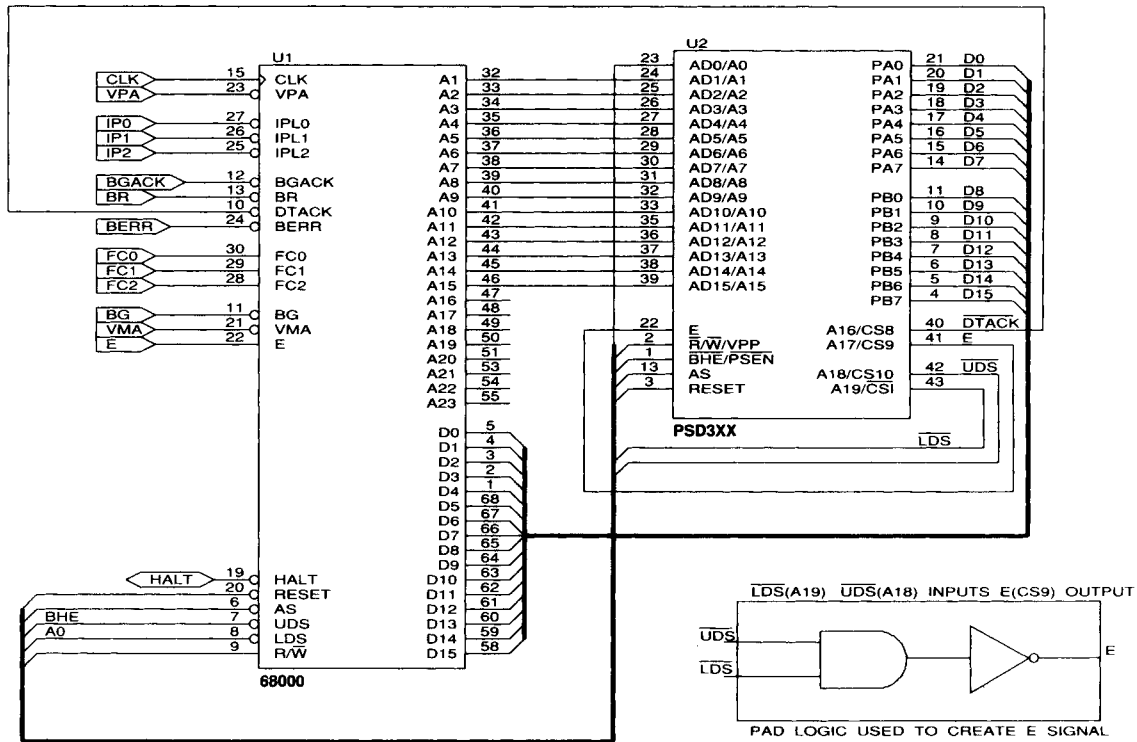


Figure 19

donnés figures 17, 18 et 19. Voilà, c'est tout pour aujourd'hui en ce qui concerne les grandes généralités de cette famille de composants, mais en attendant la suite du mois prochain commencez donc dès à présent à

dessiner avec beaucoup de soins les «mappings mémoires» de vos conceptions en cours. Vous en aurez besoin le mois prochain. A bientôt donc.

Dominique PARET

**RESERVEZ VOTRE CADEAU**

**electronique**

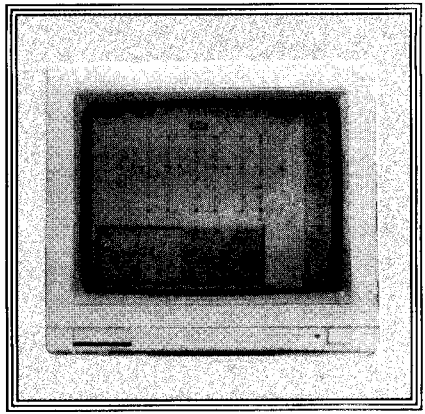
*vous offre une disquette*

**voir notre offre spéciale d'abonnement**

**CAO** sur PC/AT et compatibles **PRIX A PARTIR DE 1290 F HT** pour la version «personal» **COMMENT ECONOMISER SANS COMPROMIS ?**

**«ELECTRONICS WORK BENCH»**  
Le laboratoire d'électronique sur disquette

- Logiciel permettant de réaliser, simuler et essayer des circuits analogiques et numériques.
- Facile d'utilisation (icônes), et apprentissage rapide.
- Idéal pour apprendre l'électronique.
- Idéal pour l'expérimentation et le prototypage.
- 3 versions pour PC au choix ; personal, professionnelle, et professionnelle plus.
- Version Macintosh Plus (ou mieux), en monochrome seulement.
- Minimum de 286 avec 640k, DD, EGA/VGA, et souris recommandés pour les versions professionnelles.



Veillez m'envoyer la documentation et le tarif complet du logiciel «Electronics Workbench»

NOM : \_\_\_\_\_ Entreprise : \_\_\_\_\_

Adresse : \_\_\_\_\_

Tél. : \_\_\_\_\_

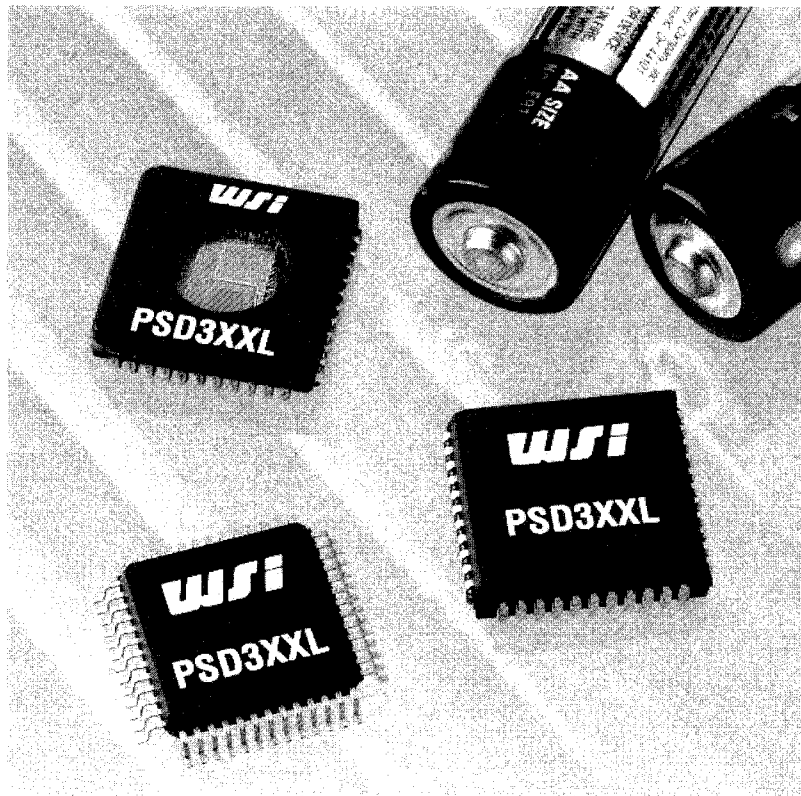
# ■ Périphériques programmables pour microcontrôleurs (2)

*Le mois dernier, nous vous avons présenté les grandes lignes d'une famille récente de «périphériques programmables pour microcontrôleurs» et nous vous avons indiqué brièvement leurs applications générales.*

*La plupart du temps, après un intérêt «d'estime» les utilisateurs potentiels s'arrêtent là en se disant que cela est très intéressant et qu'ils verront plus tard.*

*La raison ? Elle est excessivement simple ! Ils ne manquent certainement pas de courage, mais de temps pour apprendre en détail comment les utiliser.*

*Il est vrai que souvent les applications simples, de principe, standards, sont présentées dans les «hand books» mais comme par un fait exprès ils ne trouvent jamais (et pour cause) «leurs» cas particuliers qui ne sont pas spécialement plus compliqués, mais qui sont différents !*



Bref, pendant ce temps-là l'eau coule sous les ponts, et les nouveaux projets repartent pour un tour de plus avec des composants anciens, connus et parfois moins pratiques.

En revenant à cette gamme de produits «PSD 3xx» on peut dire en très raccourci qu'elle est destinée à deux grands types d'utilisation :

- simplifier fortement l'existant ;
- concevoir du «nouveau» plus compact et plus souple, avec tous les avantages que nous avons cités lors de notre précédent article.

Concevoir du «nouveau» quand on ne sait pas comment ça fonctionne tient de la folie douce ! Aussi, afin de vous expliquer comment concevoir du «nouveau», nous allons, dans cet article, à l'aide d'un exemple existant (choisi pour sa complexité bien représentative !), vous expliquer en détails et le plus pédagogiquement possible, comment éviter les pièges quotidiens dans

lesquels vous seriez sensés vous engouffrer.

Une fois que vous aurez compris comment éviter toutes ces embûches, vous serez aptes à concevoir facilement vos applications qui seront certainement très proches de celle que nous vous proposerons dans le prochain article servant de premières conclusions à l'emploi de ces composants.

## **L'exemple choisi**

La **figure 1** présente l'exemple que nous avons retenu. On vous l'accorde, il est un peu vicieux, mais sinon où serait le plaisir, n'est-ce pas ?

C'est un schéma «professionnel» bien «conventionnel».

Il comprend :

- un microcontrôleur famille 80C 51 (le 80C 32 pour être plus précis, 256 octets de RAM) ;
- de la mémoire externe RAM, EPROM et le circuit de latch d'adresses habituel



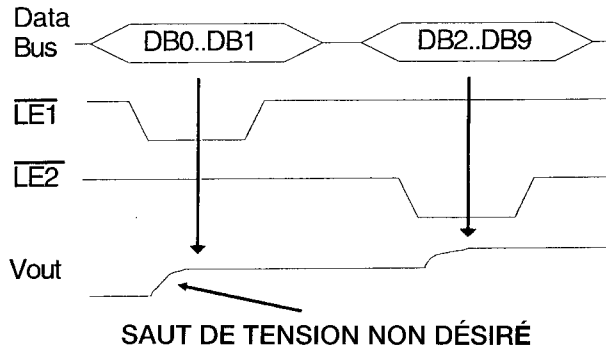
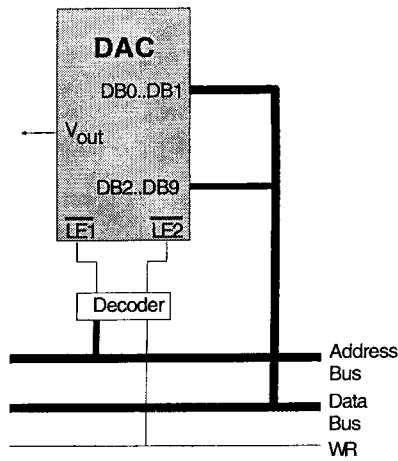


Figure 4 : contrôle du convertisseur numérique/analogique.

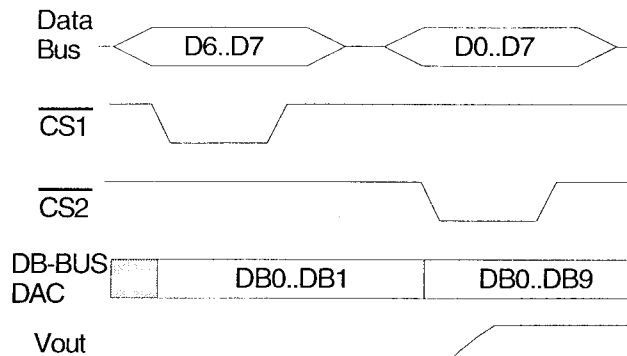
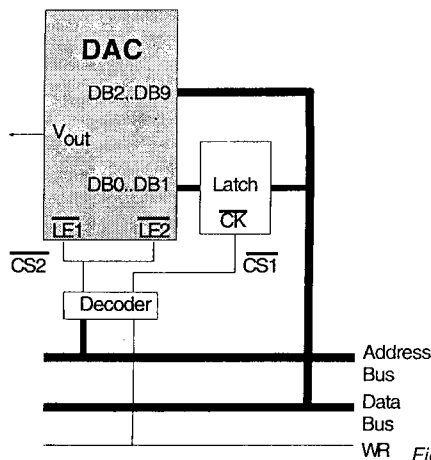


Figure 5 : version améliorée du contrôle.

Bien évidemment ce choix est personnel et libre à vous de placer ce logiciel où vous le désirez mais en vous rappelant là où vous l'avez mis !

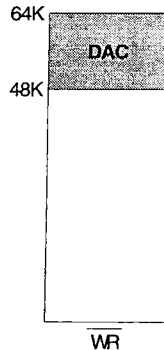
### Le convertisseur rapide sur 10 bits

Ce convertisseur est de très loin bien plus sournois !

Le bus de données du microcontrôleur ne possède que 8 bits et comme l'indique la figure 1, nous sommes obligés d'envoyer successivement sur le bus les données correspondant à DB0 et DB1 en les lachant via LE1 d'une part puis DB2 à DB9 en les lachant via LE2 d'autre part. Or ceci pose problème !

En effet, le convertisseur aurait aimé que vous lui «présentassiez» non pas 10 bits en deux temps (2 puis 8), mais 10 d'un coup. Pourquoi ?

C'est bien simple. Pour lui les deux premiers bits reçus (DB0 et DB1) représentent une valeur possible dans son arsenal de valeurs numériques à convertir, ce qu'il se fait un plaisir d'effectuer



- A0=0: Latch addressed
- A0=1: DAC addressed; new value for Vout

Figure 6 : occupation mémoire du DAC.



et, catastrophe, cela produit un palier de tension non désiré sur la tension de sortie analogique du convertisseur ! (voir **figure 4**). Pour annuler cet effet, il est nécessaire de réaliser (en hard pour l'instant) un retard au chargement du premier octet à l'aide de latches extérieurs (74 HCT 74) et de portes (74 HCT 04) pour assurer la coïncidence temporelle de présentation des bits de DB0 à DB9 aux entrées du convertisseur (voir **figure 5**).

Dernier point. De la même façon que pour l'UART, il est nécessaire de positionner le logiciel afférant

dans une zone mémoire connue, cela aide parfois !

C'est ce qui est notamment réalisé à l'aide de A0 dans la zone de 48K à 64K en mode d'écriture uniquement, **figure 6**.

### L'espace mémoire RAM et EPROM

La **figure 7** indique l'architecture retenue (désirée) pour les espaces EPROM et RAM «officiels», ceux destinés à des usages spécifiques venant de vous être présentés précédemment.

Comme vous pouvez le remarquer sur le schéma, il a été déci-

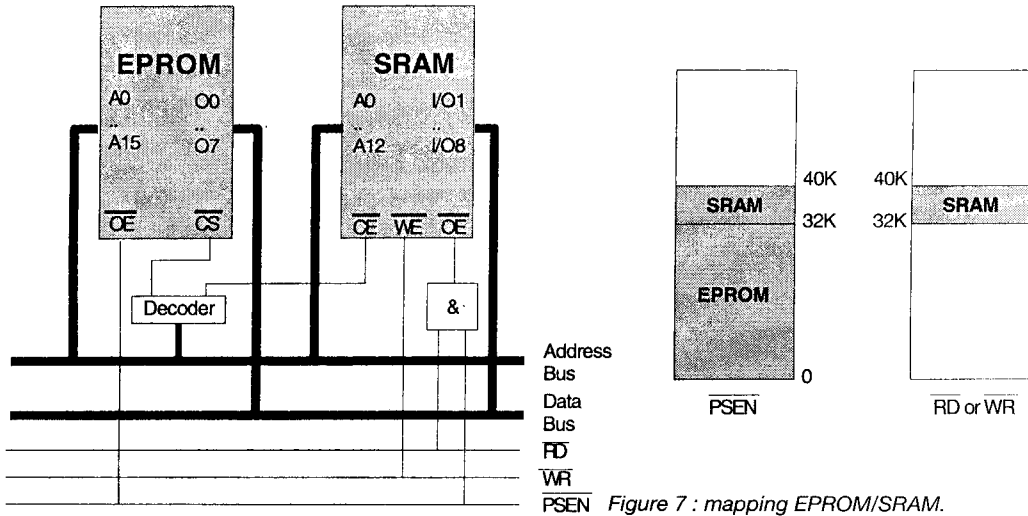


Figure 7 : mapping EPROM/ SRAM.

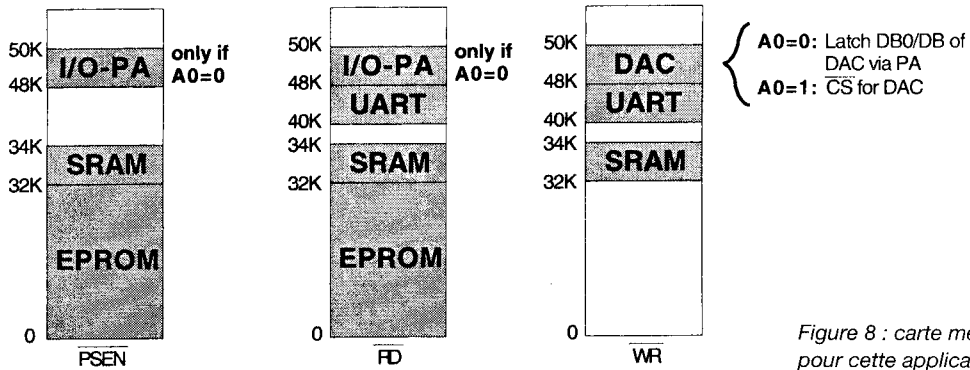


Figure 8 : carte mémoire du PSD301 pour cette application.

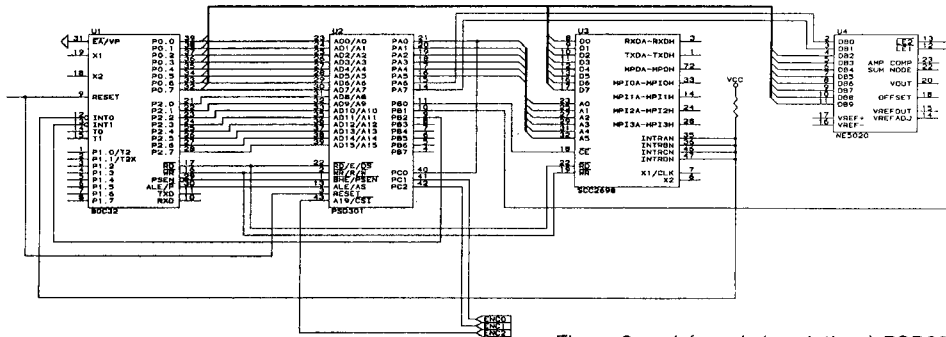


Figure 9 : schéma de la solution à PSD301.

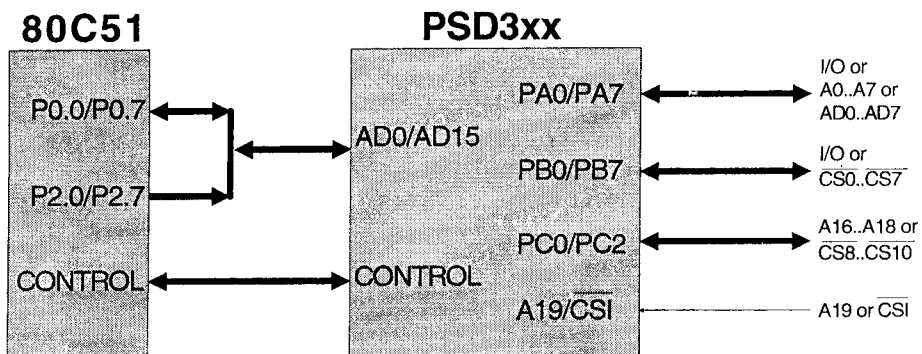


Figure 10 : configuration avec un 80C51.

dé de pouvoir disposer (à des fins par exemple de téléchargement) un «bout» de code en RAM, ce qui nécessite que celle-ci puisse être aussi activée via le PSEN (program store enable). Comme d'habitude, cela oblige donc à réaliser un ET entre le Read et le PSEN au niveau de l'accès RAM.

De plus, pour assurer le bon accès aux champs d'adresses souhaités, il est nécessaire de disposer d'un décodage des adresses à l'aide de portes (74 HCT 32 et 04).

#### **Dernière remarque**

Non liée directement au mapping, admirez la petite logique de décodage d'interruption venant signaler au microcontrôleur toutes catastrophes externes via la broche INT1. A ce sujet, toutes celles provoquées par l'UART se présenteront au microcontrôleur via la broche INTO.

#### **En résumé du mapping**

Les différents paragraphes précédents permettent de réaliser le mapping complet de l'exemple choisi et de savoir quels seront les différents points sensibles à résoudre lors du passage «composants discrets» au «PSD 3xx».

La globalité de ce mapping est donnée **figure 8** où vous remarquerez quelques petits changements du fait d'une répartition plus harmonieuse de l'espace RAM.

A ce stade on peut estimer que l'analyse du schéma est terminée ou bien n'aurait jamais dû exister car déjà existante dès la conception du projet !

#### **Résultats**

La solution de notre problème est donnée **figure 9**.

On ne peut faire plus simple ! Il ne reste que l'essentiel.

«Adieu EPROM, RAM, latches, décodeurs, portes...», seul un PSD 3xx a fleuri.

De plus nous avons réussi à relier toutes les broches entre elles.

Comment est-ce possible ? Deux figures pour expliquer cela :

La première, **figure 10**, vous rappelle le schéma de principe de liaison entre un microcontrôleur de la famille 80C 51 et un PSD 3xx.

L'important de cette figure consiste à bien s'imprégner de ce que sont sensés sortir les ports PA, PB, PC du PSD 3xx.

PA0 à PA7 : des I/O, ou des A0..A7, ou des AD0..AD7.

PB0 à PB7 : des I/O, ou des CS0..CS7 (chips select).

PC0 à PC2 : A16 à A18, ou des CS8..CS10 (chips select).

## **LE POURQUOI DU COMMENT**

### **La liaison microcontrôleur PSD 3xx**

C'est simple ! En fil à fil, les 8 bits du port P0 du microcontrôleur d'adresses/données sont reliés aux ports AD0 à AD7 du PSD 3xx. De même pour les bits du port P2 «d'adresses hautes» du microcontrôleur reliés aux AD8 à AD15 du PSD 3xx.

Il en est de même pour les signaux de services RD, W, ALE, PSEN et le Reset du 8x Cxxx que l'on relie «directement» aux broches de même nom du PSD 3xx (! sic et resic !) Ah si toutes ces broches avaient été face à face, ç'aurait pu être super. Enfin, nul n'est parfait. Bref avec le plus de l'alimentation et les «2» moins de «masse», nous avons quand même achevé  $((8+8) + (4+1) + 3) = 24$  broches sur les 44 du circuit.

Passons maintenant aux broches spécifiques et commençons à nouveau par l'UART.

#### **L'UART**

Les données D0 à D7 sur huit bits sont celles provenant du port 0 du microcontrôleur.

Les signaux d'adressage de l'UART (A0 à A5) sont fournis par le PSD 3xx par les ports PA0 à PA5 qui ont eu la gentillesse (interne) de latcher les signaux d'adressage fournis par le bus multiplexé temporellement du microcontrôleur (entre parenthèses, le PSD 3xx était prévu pour cela !) et qui fonctionne en semi-«transparence».

Le Chip Enable du SCC 2698 est commandé via le port PB0.

Pour commander ce «CE», la broche PB0 sera programmée en interne afin de satisfaire l'équation logique réalisant l'adressage du composant souhaité par le mapping général de l'application.

### **Le convertisseur D/A rapide 10 bits**

Tout devient simple ici aussi. Ouf ! Toutes les astuces «hardware» des paragraphes précédents s'estompent en un coup de baguette magique à l'aide des PAD internes qui permettent, latches internes aidant, de pouvoir présenter non pas 2 bits puis 8, mais un bus de 10 bits en une seule fois à l'aide des 8 bits provenant du microcontrôleur et de 2 déjà lachés sur les ports PA6 et PA7. Bien sûr, ils ne sont pas tout à fait simultanés, mais ce qui est important, c'est qu'ils soient présents simultanément

à un instant déterminé qui est celui où les signaux LE1 et LE2 seront appliqués au circuit convertisseur.

A votre avis qui va réaliser ce petit miracle ? Le «PAD B» du PSD 3xx en utilisant les signaux internes adéquats et dont le résultat «logique» sera disponible sur la broche PB1 et le tour sera joué.

Cela tient de la conjonction de la logique programmable et de la logique microcontrôlée.

#### **Les interruptions**

Ici aussi on a mis toutes les portes à la «porte». Quel comble ! Il y a de quoi sortir de ses gonds !

En effet, le PLA interne ne fait qu'une bouchée de la malheureuse équation logique comportant ces trois ou quatre portes. Les différentes sources d'interruptions rentrent sur PC1, PC2 et A19 et, après traitement, nous avons décidé de faire ressortir le résultat en PB2 pour attaquer l'entrée INT1 du microcontrôleur.

Le prochain article traitant en détail de la programmation des composants vous exposera comment nous avons réussi ce merveilleux prodige d'une complexité époustouflante (qui consiste à cocher au moins une ou deux cases dans le logiciel de support de conception !)

Bref, un travail intense ! (chut, il ne faut rien dire à votre patron).

#### **L'organisation des plans mémoire**

Quelles mémoires ? Mais où sont-elles donc passées ? On ne les voit plus sur le schéma.

Rassurez-vous, elles sont toujours présentes mais intégrées dans le PSD 3xx et leurs mappings avec !

#### **En conclusion**

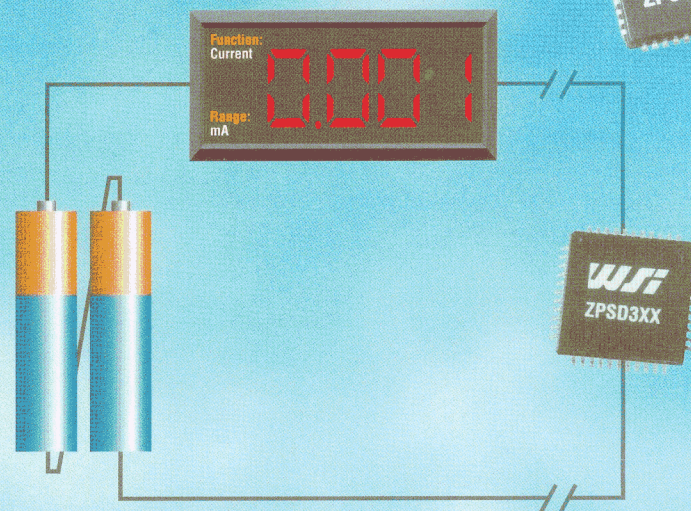
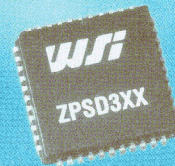
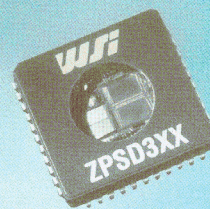
Voici terminée la mise en œuvre «papier» de cette réalisation d'application comportant des circuits périphériques programmables pour microcontrôleurs de type PSD 3xx.

Nous vous donnons à présent rendez-vous le mois prochain pour conclure ce sujet en vous exposant, d'une part, une réalisation pratique générale facilement personnalisable en fonction de vos propres projets et, d'autre part, nous en profiterons pour vous décrire les logiciels associés d'aide à la conception ainsi que les outils permettant de réaliser la programmation proprement dite des composants.

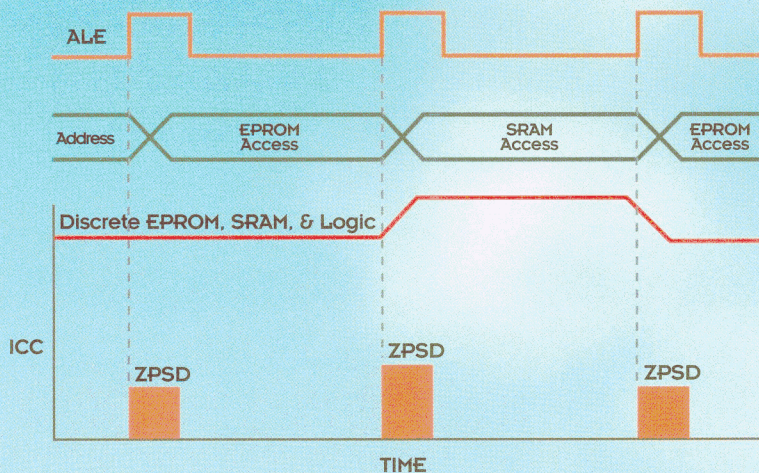
**Dominique PARET**

# ZPSD3XX Zero-Power Programmable MCU Peripherals

The ZPSD3XX Zero-Power programmable MCU peripheral family is ideal for low-power portable applications. Included on a single chip is up to 2 Mb of EPROM, 16 Kb SRAM, programmable memory map address decoder, "no glue logic" MCU interface, programmable I/O ports, low-power options, 70-ns performance and more! And all operating at virtually no power!



## ZPSD Power vs. Discrete Implementation

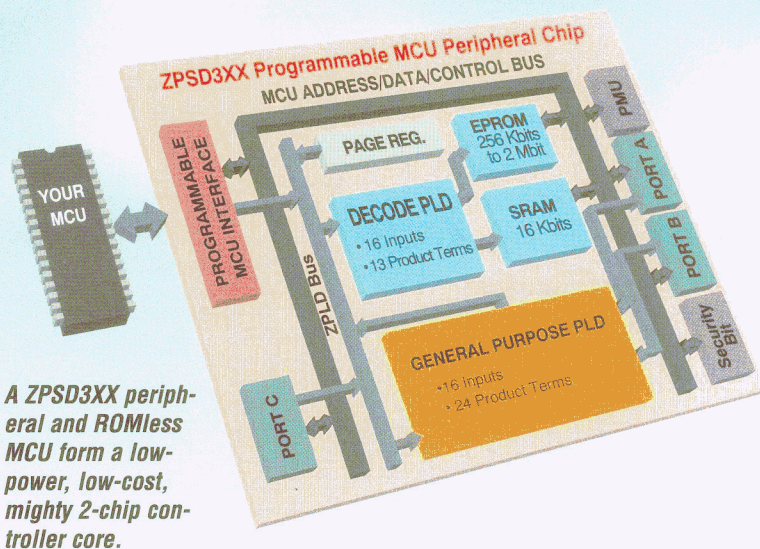


The 2.7 volt ZPSDs draw only 1 microamp in standby and just 0.4 mA/MHz bus frequency. The 5-volt ZPSDs take only 10 microamps in standby and 0.8 mA/MHz bus frequency. For example, the operating current of a 5-volt ZPSD when used with an 8031 running a 4 MHz oscillator is only 533 microamps. This represents an 8-to-16-fold improvement over a discrete implementation!

The ZPSD is always automatically in standby mode except for the tiny percentage of time in each cycle when it is fetching and latching data. The payoff is low, low operating power and much longer battery life.

Use windowed erasable ZPSDs for quick prototyping. Then move to OTP devices for production. For code-stable, penny-picing, high-volume production, use our new mask-programmable ZPSDs. WSI, Inc., and American Microsystems, Inc. alternately source masked-programmable ZPSDs.

The new ZPSDs can extend cellular phone standby time by several hours. The 2.7 Volt ZPSDs enable you to get rid of a battery and make smaller, lighter portable products. Now you can use telephone line or serial port current to power your products. Use a low power ZPSD3XX peripheral whenever you need more resources than provided by your MCU or whenever you need to reduce the size and power consumption of your products.



*A ZPSD3XX peripheral and ROMless MCU form a low-power, low-cost, mighty 2-chip controller core.*



47280 Kato Road, Fremont, CA 94538 • Phone: (510) 656-5400 • Fax: (510) 657-8495  
800-TEAM-WSI (800)-832-6974 • In California 800-562-6363

# PSDsoft Development Tools

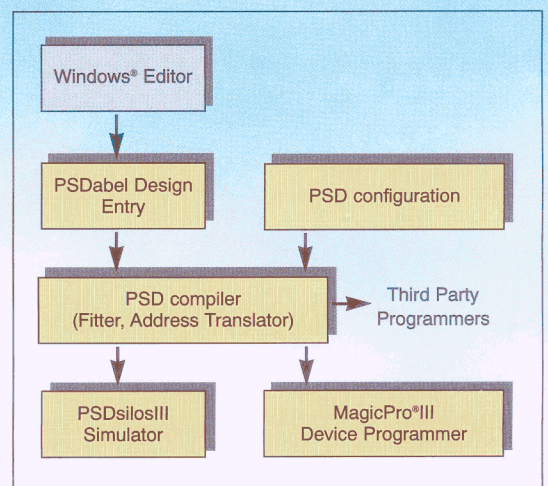


PSDsoft Development tools for the ZPSD3XX family operate on a 386/486 or Pentium® PC under Microsoft® Windows® 3.X. WSI has seamlessly interfaced industry standard software design tools and applied them to the Windows framework to enable an easy to use "point and click" design methodology.

By following a logical step-by-step design procedure, a ZPSD3XX device can be quickly configured and programmed. PSDsoft pricing starts at \$495.

The five major software modules included in Windows-based PSDsoft are:

- **PSDabel™**—the WSI Windows version of the widely popular Data I/O ABEL® logic design software.
- **PSD configuration**—specifies the MCU bus interface type and I/O port pin configurations.
- **PSD compiler**—the Fitter fits the logic and I/O functions into the ZPSD3XX device of your choice. The Address Translator merges your code file and generates an object file which is used by the PSDsilosIII™ simulator or PSD programmer software. The file includes chip configuration and fuse map information along with EPROM program code.
- **PSDsilosIII™ simulator**—the easy-to-use PSDsilosIII™ is the WSI version of Simucad's SILOS® simulation software. It provides full chip-level simulation and comprehensive node analysis of the ZPSD3XX.
- **PSD programmer**—used for programming a ZPSD3XX device on the WSI MagicPro®III PC-compatible programmer.



## The MagicPro®III programmer

is an engineering development and prototyping tool and is used to program any ZPSD3XX device. The programming menu in PSDsoft simplifies programming a ZPSD3XX device on the MagicPro®III. Pricing begins at \$895.

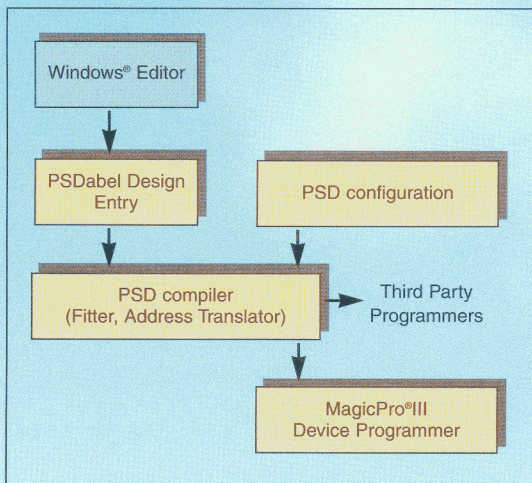
# PSDsoft Lite Development Tools



PSDsoft Lite enables the user to rapidly configure and program a PSD3XX or ZPSD3XX device by following a logical step-by-step "point and click" design procedure. PSDsoft Lite development software for PSD3XX and ZPSD3XX products operates on a 386/486 or Pentium® PC under Microsoft® Windows® 3.X or Windows 95®.

The four major software modules contained in PSDsoft Lite are:

- **PSDabel™**—the WSI Windows version of the popular Data I/O ABEL® logic design software.
- **PSD configuration**—specifies the MCU bus interface type and I/O port pin configurations.
- **PSD compiler**—the Fitter fits the I/O functions and logic into the particular PSD3XX or ZPSD3XX product you've selected. The Address Translator merges your code file and produces an object file used by the PSD programmer software. The PSD compiler file includes chip configuration and fuse map information along with EPROM program code.
- **PSD programmer**—used for programming a PSD3XX or ZPSD3XX device on the WSI MagicPro®III PC-compatible programmer.



**The MagicPro®III programmer** is an engineering prototype and product development tool and is used to program any PSD3XX or ZPSD3XX device regardless of package. The PSDsoft Lite programming menu greatly simplifies the programming function. The MagicPro®III is priced at \$895.

Purchase a PSDsoft Lite (WSI Part Number WS7002L) and get started on your design right away. Have your credit card handy and call any of the following USA distributors for instant service.

Arrow 800-833-3557  
 Avnet 800-332-8638  
 Wyle 800-414-4144



# ZPSD3XX Zero-Power Programmable MCU Peripherals

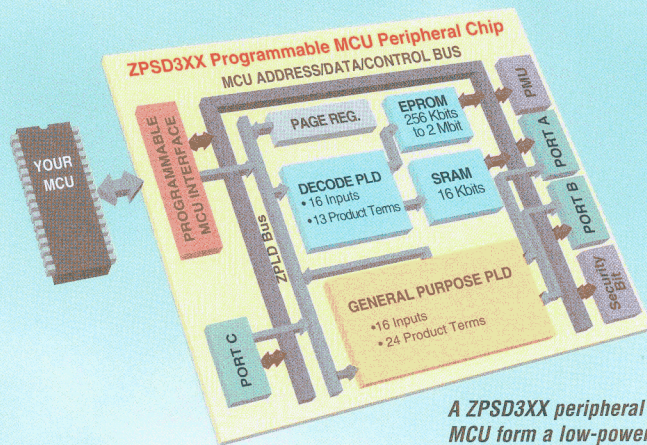
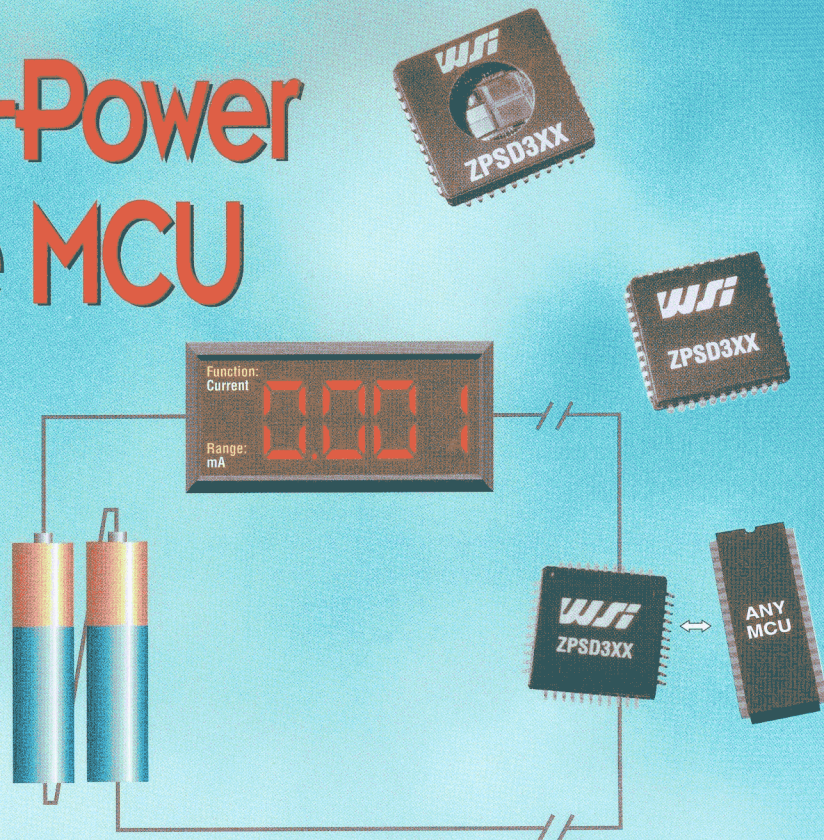
ZPSD3XX Zero-Power programmable MCU peripherals are ideal for low-power portable applications. Included on a single chip is up to 2 Mb of EPROM, 16 Kb SRAM, programmable memory map address decoder, "no glue logic" MCU interface, programmable I/O ports, low-power options, 70-ns performance and more! And all operating at virtually no power!

The 2.7 volt ZPSDs draw only 1 microamp in standby and just 0.4 mA/MHz bus frequency. The 5-volt ZPSDs take only 10 microamps in standby and 0.8 mA/MHz bus frequency. For example, operating current of the 5-volt ZPSD is only 533 microamps when used with an 8031 running a 4 MHz oscillator. This represents an 8-to-16-fold improvement over a discrete solution!

The ZPSD is always automatically in standby mode except for the tiny percentage of time in each cycle when it is fetching and latching data. This low power operation will not slow down your system and requires no special hardware or software. The payoff is low, low operating power and much longer battery life.

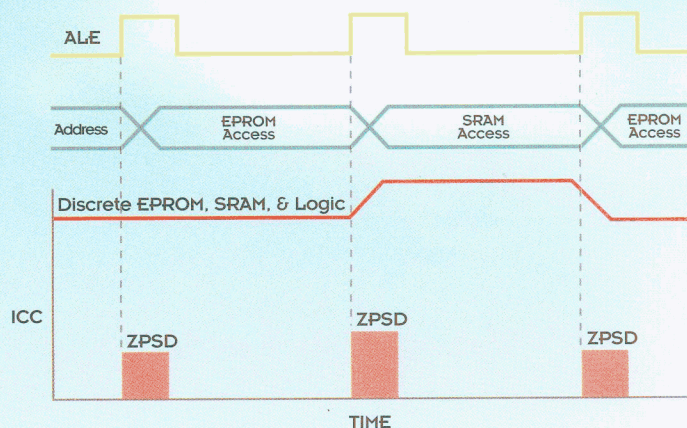
Use windowed erasable ZPSDs for quick proto-typing. Then move to OTP devices for production. For code-stable, penny-pinching, high-volume production, use our new mask-programmable ZPSDs.

Low power ZPSDs can extend cellular phone standby time by several hours. The 2.7 Volt ZPSDs enable you to get rid of a battery and make smaller, lighter portable products. Now you can use telephone line or even serial port current to power your products. Use a low power ZPSD3XX peripheral whenever you need more resources than provided by your MCU, or whenever you need to reduce the size and power consumption of your products.



A ZPSD3XX peripheral and ROMless MCU form a low-power, low-cost, mighty 2-chip controller core.

## ZPSD Power vs. Discrete Implementation



47280 Kato Road, Fremont, CA 94538 • Phone: (510) 656-5400 •  
 Fax: (510) 657-8495 • 800-TEAM-WSI (800)-832-6974 •  
 In California 800-562-6363 • Website <http://www.wsipsd.com>