iCC 1997

4th international CAN Conference

in Berlin (Germany)

Sponsored by

Motorola Semiconductor
NEC Electronics (Europe)
Siemens Semiconductors

Organized by

CAN in Automation (CiA)
international users and manufacturers group
Am Weichselgarten 26
D-91058 Erlangen
Phone +49-9131-69086-0
Fax +49-9131-69086-79
Email:headquarters@can-cia.de
URL: http://www.can-cia.de

Dipl.-Ing. Gerhard Hausmann
Assistant Manager Sales Automotive
NEC Electronics (Germany) GmbH

Dipl.-Ing. Edgar Gebing
Application Segment Support Automotive
NEC Electronics (Europe) GmbH

The realisation of specific automotive applications with “Full” CAN functionality at “Basic” CAN cost on highly integrated 8-Bit microcontroller of NEC’s 78K/0 family.

CAN today is the most widespread serial communication protocol in automotive applications. Coming from the high speed, high performance applications like motor management, where CAN is fairly well established in almost all major car makers around the globe, it has now conquered the wide field of comfort and convenience electronics as well as driver information systems. Although numerous silicon manufacturers have introduced a wide range of integrated microcontrollers with on-chip CAN, there was still lack of Application Specific Standard Products (ASSP) with CAN. NEC, an early pioneer of CAN activities, has established a complete new family of highly integrated microcontrollers specifically for the area of automotive instrumentation clusters. The new “Direct storage CAN” (DCAN) module of NEC ideally combines a CAN 2.0b extended frame compliant with “Full” CAN functionality at “Basic” CAN cost. The integration of highly competitive and application specific peripherals on NEC’s powerful 78K/0 family in combination with Flash memory technology offers new perspectives and opportunities for system solutions at excellent price performance ratio.

Introduction

The number of applications in industrial and automotive electronics which require CAN are growing rapidly. In line with this trend there is an increasing demand for microcontrollers with application specific peripherals and on chip CAN functionality. Based on the powerful and well established 78K/0 8-bit and V850 32-bit RISC microcontroller families NEC offers highly integrated application specific embedded solutions comprising E^2PROM, FLASH technology, intelligent peripherals and a high performance DCAN interface.

A major challenge for the integration of CAN on silicon is to find an optimum compromise between the performance of the CAN interface, i.e. minimum CPU load for communication, and cost efficiency of the implementation. The flexible DCAN concept from NEC combines both requirements in an ideal way.

This paper describes the functionality of the DCAN and the µPD78(F)0948, which is a member of a new 78K/0 family branch with on-chip DCAN.

DCAN (Direct storage CAN)

The DCAN Concept

The target of the DCAN concept was to offer an optimum compromise between a minimum CPU load for the communication and a cost-effective implementation.

Due to the broadcast nature of CAN each message will be provided to each participant in the network. Analysis on low end CAN implementations have shown that quite some CPU load may occur to select between desired and undesired CAN messages in case that the acceptance filtering of the CAN interface is not sufficient. This is due to the fact, that a node has normally to receive high priority as well as low priority messages, i.e. the receive messages of a certain node are distributed over the identifier space. Simple acceptance filtering normally does not allow to specify a very precise filter characteristics. As a result a lot of undesired messages may be received by the CAN interface and have to be post-filtered by the CPU. Moreover it might be difficult for a designer to consider the required CPU performance in the design phase of an ECU, if in a later stage the number of bus messages increase due to a growing network.

In ideal case the complete filtering is done by the CAN interface and the messages received by the node are represented and updated direct in dedicated mailboxes. Due to the fact that the messages are already sorted by CAN there is no need for the CPU to transfer the data into a final application RAM location.

In a low end CAN implementation the CPU has to do the transfer which has also to be considered with respect to the real time performance of the whole system (Interrupt latency), especially in a high speed CAN network.

On the other hand the handling of transmit messages is anyhow related to the CPU activity of the application program, i.e. the contents of transmit messages has to be generated by the CPU.
However, in order to avoid a complex management of transmit messages by the CPU, it is desirable to have more than one transmit buffer. The reason is that the application might generate messages with different priorities faster than they can be transmitted in subsequent order.

The DCAN represents an ideal solution providing best hardware support for most autonomous CAN communication based on a minimum circuit design.

Transmission is supported by two independent transmit buffers with easy priority control. The receive path provides nearly “Full-CAN” performance with up to 16 mailboxes in the communication RAM.

The implementation of these powerful features with the condition of a minimum circuit size were possible by a separation of the transmit and receive path and the minimisation of storage area for temporary data.

DCAN Features

- Implementation of CAN 2.0B active
  - Reception and transmission of standard and extended frame format messages.
- Bitrates up to 500 Kbaud.
- Up to 16 receive mailboxes
  - Configurable number of receive mailboxes between 0 and 16
  - Receive messages are directly stored into the final RAM location
    (**D**irect storage **CAN**: Full-CAN principle)
  - All mailboxes can handle 11 or 29 bit identifier
  - Standard and Extended frame format messages can be mixed
- Acceptance filters can be assigned to two mailboxes for multi message reception
  - Acceptance filters support full 11/29bit identifier
  - Global Mask function
- Two independent transmit buffer for transmission of standard/extended frame format message
- CPU and DCAN share same RAM block of maximum 288 bytes
  - Cycle steal DMA controls access of CPU and CAN unit
  - Unused communication RAM can be used as application RAM
    - Less than 16 receive mailboxes
    - Standard frame messages
    - Messages with less than 8 data bytes
  - Very easy and secure data handling between CPU and DCAN
- Interrupt support for transmit, receive and error signalling
  - Interrupt can be enabled or disabled for each individual mailbox

- Readable error counters
- Redefinition function
    - A mailbox can be put into off line state to be set up for another message
    - All other mailboxes remain under communication
- Bus listening mode (no acknowledge, no active error flags)
    - Valid protocol activity detection
- Automatic baud rate detection
- Time stamp and global time system support
- Two power save modes: Sleep and Stop mode
- Driver Software Library available

DCAN Functional Description

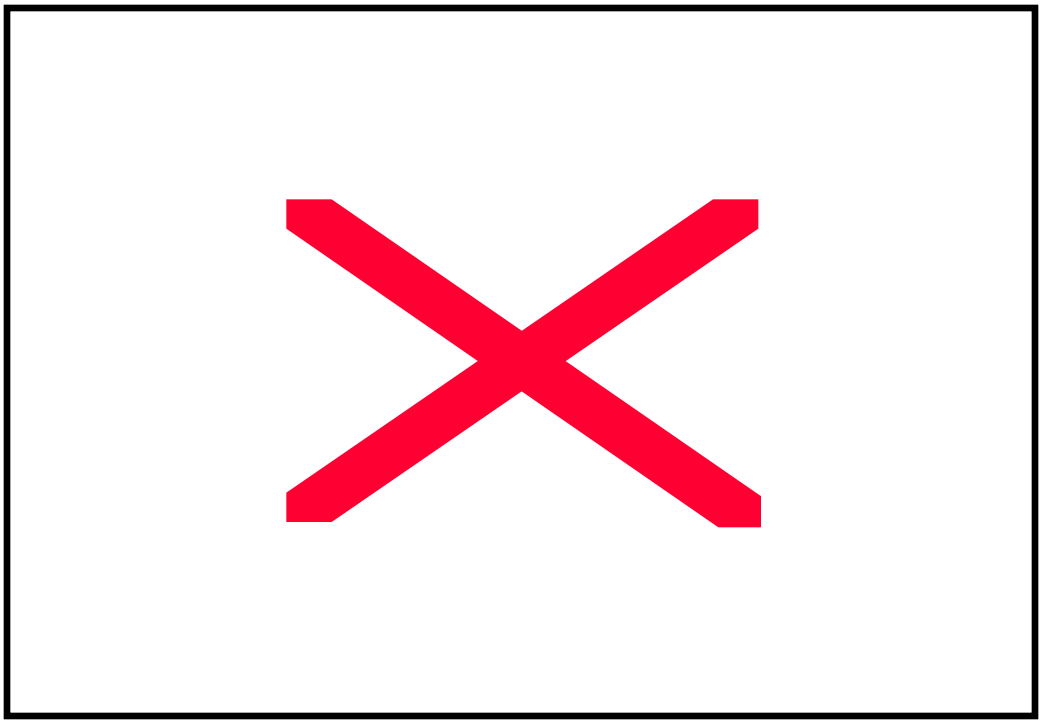

Figure 1: Block diagram of the DCAN interface

The DCAN Interface Block

The DCAN interface unit is the main block of the on chip DCAN macro and includes the CAN protocol block, an interface management unit and a memory access engine. The CAN protocol block comprises all functions of the transfer layer according to the Bosch specification 2.0. A feature of this block is the so-called listening mode, which allows to monitor the bus activities and to display valid protocol activities by a status bit to the CPU. In this mode, the interface listens to the CAN bus without active participation in the CAN communication, i.e. no acknowledge and error signalling. With this feature the CPU can perform automatic baudrate detection without corrupting the external CAN bus communication.

To support a global time base and generate time stamps of received messages the DCAN interface block provides trigger signals, which are connected to some capture input of a timer unit. With the new on chip DCAN concept NEC continues to support these kind of real time support features, which were introduced by NEC in the early days of CAN implementations [1?].

The DCAN is designed for easy connection to standard transceivers, which are offered by several semiconductor suppliers, using two terminals (CTXD and CRXD). Alternatively to the internal clock source an external CAN clock input (CCLK) can be used for the DCAN interface.

The Cycle Steal DMA Control Unit

The DMA is an essential part of the on chip DCAN functionality. It can be described as a „Cycle Steal DMA“, which synchronises the memory accesses of the memory access engine of the DCAN

bus the access will be open for the DCAN and vice versa. The circuit is designed in such a way, that all DCAN operations can be performed simultaneously to normal program execution of the CPU with 125 ns cycle time without any restrictions.

The Internal Expansion RAM / DCAN RAM

The expansion RAM is used in 78K/0 standard products to expand the internal high speed RAM. As already explained before hand, a part of this RAM is used as CAN data acquisition RAM to the CPU. A flexible number of mailboxes for transmission and reception of CAN messages can be configured by setting the SFR registers of the DCAN.
The data structure of those mailboxes includes the necessary semaphores and status bits to assure consistent data exchange between the DCAN and the CPU. Figure 2 shows the outline of the DCAN memory and the data structure of a receive mailbox. The Data New bit (DN) is used to signal a message update in the mailbox and the bit MUC signals, that a memory update by the DCAN is ongoing. For a faster access of the CPU to receive messages, the DN bits of the first eight receive mailboxes are mirrored into the RMES register.

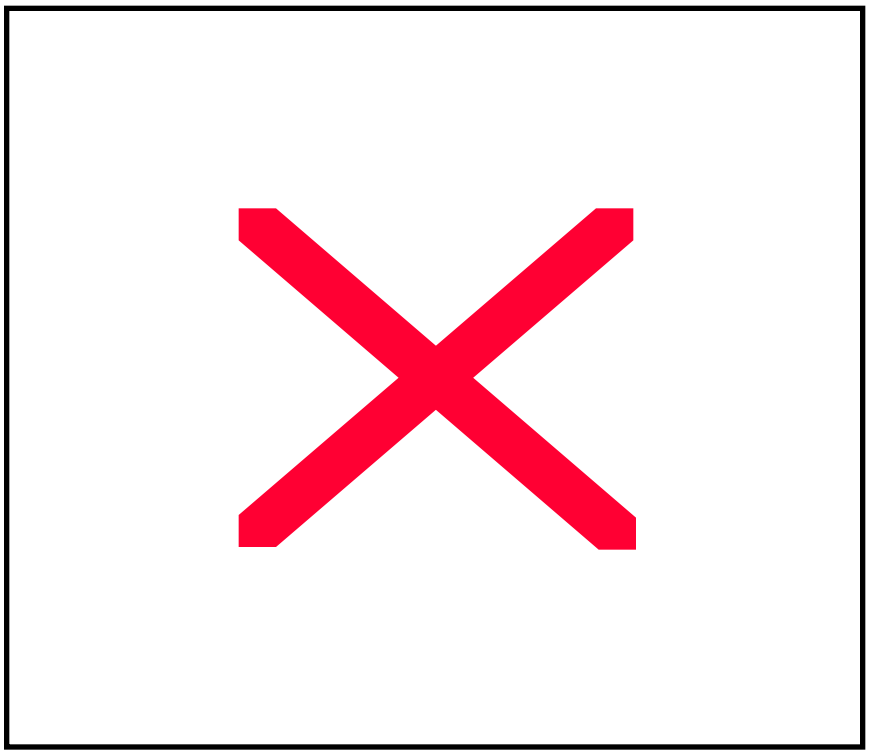

Figure 2: Memory layout of the DCAN communication RAM

Due to the fact, that up to 16 receive mailboxes can be configured, which are dedicated to unique messages, the DCAN architecture offers the performance of a so called „Full“-CAN controller on the reception path. To extend the number of receive messages to more than 16, two of the receive mailboxes can be configured as multi message buffers with the aid of 11 resp. 29 bit wide masks for acceptance filtering. In this case, up to two multi-message receive buffers and additional 12 single message mailboxes can be chosen.
Two transmit buffers support the transmission of messages. An effective priority control allows simple transmit handling by the CPU and avoids the so-called priority inversion problem [2?].
Each message buffer allocates 16 bytes of expansion RAM. Bytes, which are not used for communication by the DCAN, like

- unused bytes in each 16 byte block
- unused ID bytes when using standard instead of extended ID
- unused message data bytes
- or unused receive mailboxes or transmit buffers

can be used by the CPU to store other application data in these RAM locations.

Fully autonomous Reception of Messages

Due to the fact, that the DCAN supports a number of receive mailboxes uniquely assigned to a single CAN message, the reception path operates fully autonomously without any support from the main CPU. Thus the performance of the reception path of the DCAN is comparable to a CAN interface commonly called „Full-CAN“. Received messages will automatically be sorted into the right mailbox. No data transfer from CPU side is required to store the message data into a final RAM location.

Setting the Data New (DN) bit of the respective mailbox will show an update of the message data in a particular mailbox. To avoid permanent scanning of the DN bits within the DCAN RAM the Data New bits of the lowest 8 mailboxes are also displayed in the RMES register as a mirror of the DN-bits in the respective mailboxes. Additionally for each mailbox the receive interrupt can be enabled with the ENI-bit. This effectively supports application oriented event message handling beside the standard polling of periodic messages.

Figure 3 shows the basic CPU operations to read event messages from a receive mailbox. Figure 4 shows the basic operations and the semaphore handling when the DCAN stores a receive message.

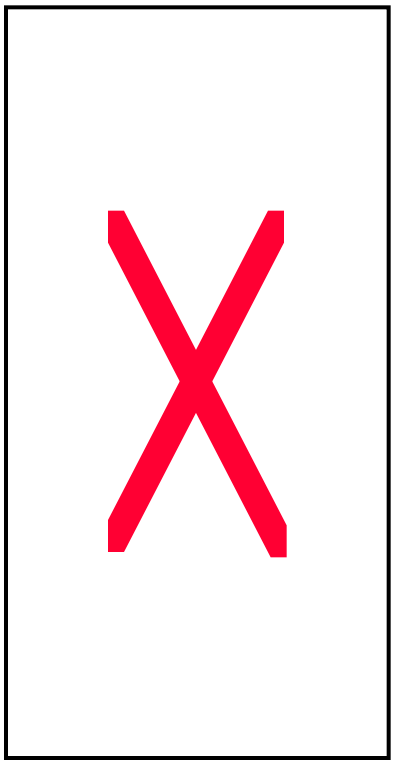

Figure 3: Reading of a received message by the CPU

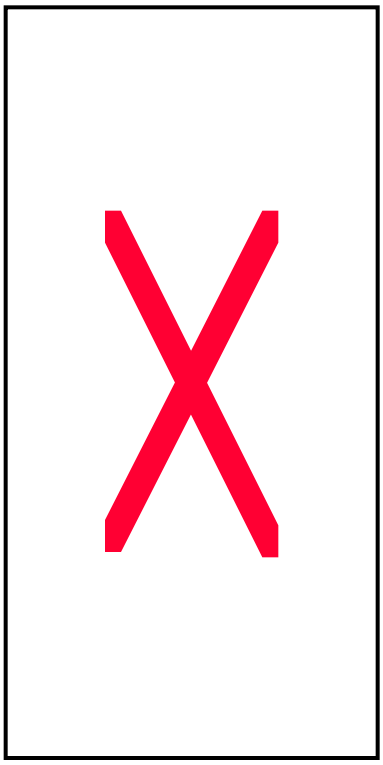

Figure 4: Handling of Semaphore Bits by DCAN for a receive message

Receive of Messages with Mask function

In addition to the reception of a unique message in a receive mailbox it is also possible to receive a group of messages within two of those mailboxes using acceptance filters with a mask as shown in Figure 2. The principle is, that the storage area of the mailbox No. 0 and 2 is used for the mask definition. Normally the next receive mailbox is assigned to the mask and operates as a multi message receive channel. Another possibility is to use a mask as a global mask for all receive mailboxes defined. Three possibilities to use the mask function are shown in the following Figures.

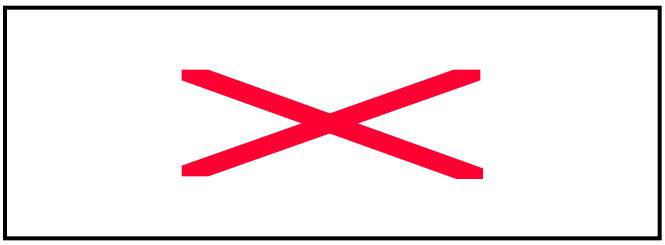

Figure 5: DCAN configuration with One Mask and up to 14 single message mailboxes

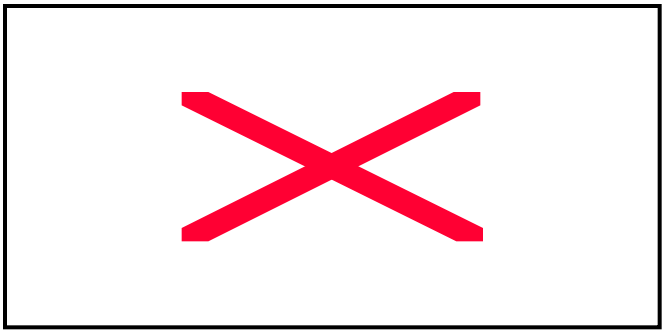

Figure 6: DCAN configuration with Two Masks and up to 12 single message mailboxes

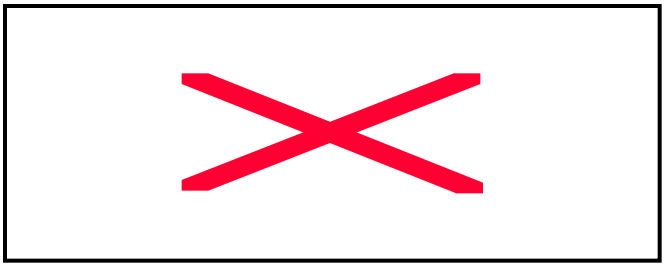

Figure 7: DCAN configuration with Global Mask function

Message Buffer 0 and Buffer 2 may be switched for masked operation. In this case the message does not hold message identifier and data, but only holds mask for identifier and RTR information for masked compares on the next higher message number. In the case of a global mask select, it keeps mask information for all higher messages (Figure 7). A mask does not store any information about identifier length. The same mask can therefor be used for both types of frames (standard and extended) during global mask operation.

The mask provides the possibility to exclude some bits of the responded identifier from the comparison process. That means each bit is ignored when the corresponding bit in the mask definition is set to one. the two diagrams in Figure 8 will show the identifier mask function.

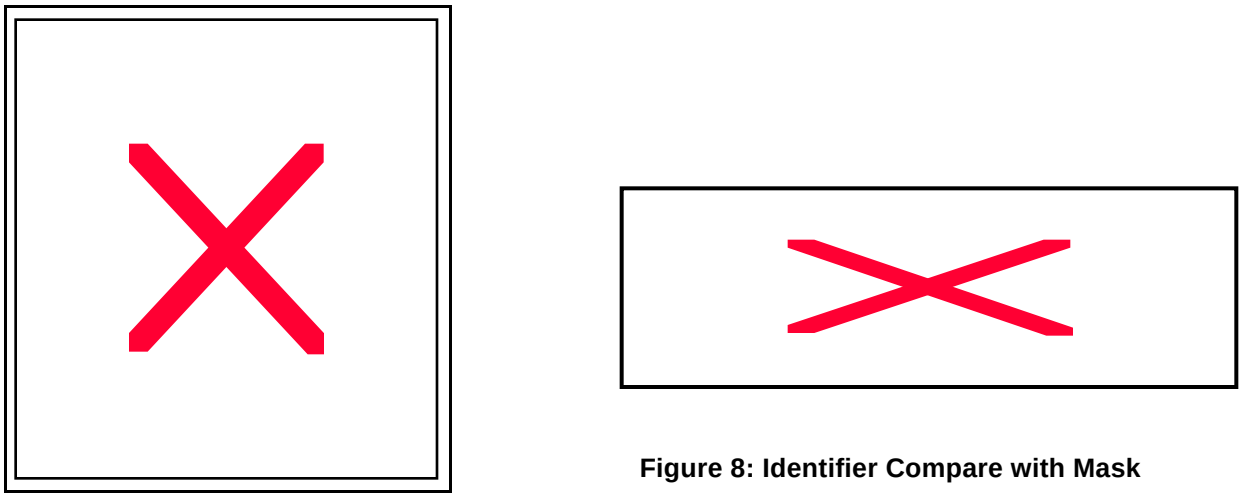

Figure 8: Identifier Compare with Mask

The mask definition allows to specify 29 bit even though the affected mailbox receives a standard frame message. The comparison of the RTR bit can also be masked. Thus it is possible to receive data and remote frames on the same multi message receive mailbox.
The message is stored from the receive shadow buffer to the right mailbox location in RAM after the whole message is received and valid.

Transmission of messages

Up to two transmit channels can be used for the transmission of CAN messages. The control bits for the transmit-channels are placed in a Special Function Register and not in the RAM area of a transmit buffer. Thus the control of the transmit buffer can be done by the CPU with a minimum of register accesses.
After the CPU has checked the availability of a free transmit buffer (TxRQn=0) the message data including data bytes, DLC and identifier (11 or 29 bits) are stored into the transmit buffer. Afterwards the transmit-request bit is set to 1. If an additional message should be transmitted, the CPU can choose the second buffer. Depending on the message identifier compared to the message in the other channel, the CPU selects the priority, which buffer should arbitrate on the bus first. Figure 9 shows the basic operations of the CPU.

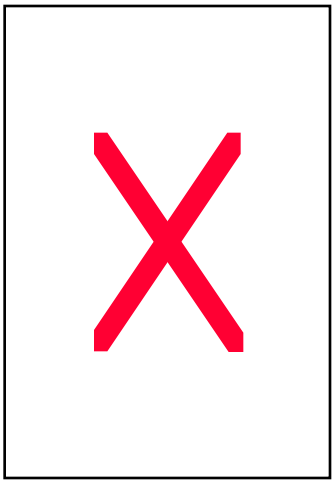

Figure 9: Transmit preparation

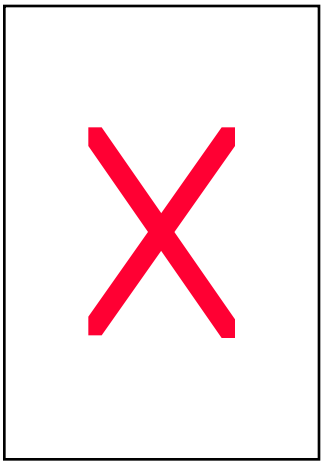

Figure 10: Transmit Abort

An abort of a requested message is not allowed at each time by the CAN specification but needed from applications. The DCAN gives a good support to abort messages without offend any rules of the DCAN protocol. The CPU can request a transmission or an abort, DCAN answers by clearing this request. How an abort is executes shows Figure 10.

An abort operation can cause different results dependent on the time it is set:
- Abort is received before the start of the arbitration for transmit
- Abort is received during the arbitration, but arbitration is lost.
- Abort is received during frame transmission, but transmission ends with an error.

=> The transmit complete flag is reset showing that the buffer was not send to other nodes.
- Abort is set during the frame transmission and transmission ends without error.

=> The transmit complete flag is set showing a successful transfer of the data before the abort gets active.

µPD78(F)0948: Application Specific Standard Product with DCAN

The µPD78(F)094X [3?][4?] is a member of a new branch of NEC´s 78K/0 8-bit microcomputer family in 0.35 µm CMOS technology with on chip CAN interface. This highly integrated Application Specific Standard Product (ASSP) was designed in the European Technology Centre (ETC) located in the European Headquarter in Düsseldorf/Germany. Figure 11 shows the block diagram of the µPD78F0948 with Flash memory, EEPROM and internal DCAN.

Features

- 60kB internal mask ROM or Flash EPROM
- 256 byte EEPROM and 2048 bytes RAM
- 64kB external memory expansion space
- Power saving modes
- Instruction execution time changeable (up to 250ns/instruction@8Mhz)
- High performance instruction set
- 16/8-bit fractional division
- 79 I/O ports
- LCD controller
- 8-channel 8-bit A/D converter
- Power failure detector
- Sound generator
- 6 Timer
- D-CAN interface
- Serial interface (2-wire, 3-wire, UART)
- 27 vector interrupts
- EMI optimised design

78K/0 8-bit microcontroller family

The 78K0 family is a well-established microcontroller family, which consist of more than 120 different products. The large variety of on-chip peripheral functions and many different memory options make the 78K/0 family suitable for nearly all kinds of applications. The product range includes a wide variety of general purpose products, but also an increasing number of Application Specific Standard Products (ASSP) like the µPD78(F)094X.
The heart of the 78K/0 family is a powerful 8/16-bit CPU. Four register banks with eight 8-bit registers can be concatenated to a 16-bit register to support 16-bit operation, e.g. 8-bit multiplication with 16-bit result or 16-bit index addressing. The 64Kb linear address space is accessible via a 16-bit wide program counter and stack pointer.
All 78K/0 microcomputers are supported by programmable derivatives. One time programmable (OTP), UV erasable EPROM and from now onwards Flash EPROM versions are pin and function compatible and provide equivalent electrical specification.

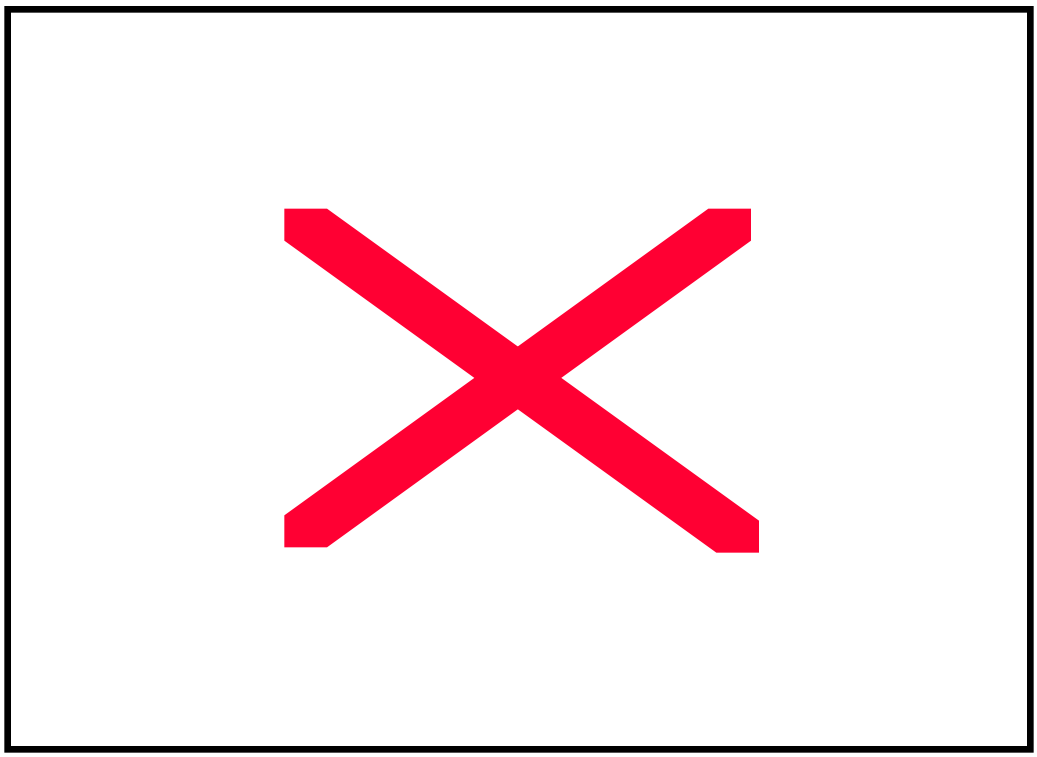

Figure 11: Blockdiagram of Microcomputer µPD78F0948 with integrated DCAN

Outlook and Summary

The µPD78(F)0948 is the first of a series of Application Specific Standard Products (ASSP) with CAN, which are designed in NEC´s European Technology Centre (ETC). This establishment of the European Technology Centre, located in the European Headquarter in Düsseldorf, allows NEC to react faster and more flexible to European customers demands. Further ASSPs based on different microcontroller cores, like the 78K/0 8-bit or the V850 32-bit RISC family, are under development for different applications. Especial for the CAN market NEC has started the design of several products based on the 78K/0 8-bit family with DCAN (Table 1), followed by products based on the V850 32-bit RISC microcontroller core with Full-CAN [5?].

Parts Name	Package	ROM	Flash	RAM	EEPROM	CAN Mailboxes	Miscellaneous
µPD780944	100 QFP	32K		1.2K	256	DCAN, 2+16	8.3 MHz/240ns + RC Sub clock, A/D + Power Fail detector, 6 Timer, PWM, 4*40 LCD, Sound Generator, External memory expansion, 2 SIO, UART
µPD780948		60K		2K	256	BasicCAN, 2	
µPD78F0948			60K	2K	256		
µPD780812	64 QFP	16K		800	256	DCAN, 2+16	8.3 MHz/240ns + RC Sub clock, Clock monitor, 6 Timer, PWM, A/D + Power Fail detector, SIO, UART
µPD780814		32K		1.3K	256	BasicCAN, 2	
µPD78F0818			60K	2K	256		
µPD780823	80 QFP	24K		1K	256	DCAN, 2+16	8.3 MHz/240ns, SIO, UART, A/D + Power Fail detector, 6 Timer, PWM, 4*30 LCD, Sound generator, Universal instrument motor driver
µPD780824		32K		1.2K	256	BasicCAN, 2	
µPD78F0828			60K	2K	256		

Table 1: Roadmap of 78K/0 products with DCAN

All new products will be offered as Mask-ROM and Flash-EPROM. The availability of flash versions ensures time efficient software development and testing, and supports ramp up production and field programming in a flexible way.

References

1? Klaus Turski: “A Global Time System for CAN Networks”, Proceedings of the 1st International CAN Conference 1994

2? Dr. K. Tindell, A. Burns: „Guaranteeing Message Latencies in CAN“, Proceedings of the 1st International CAN Conference 1994

3? NEC: „78K/0 Product Letter µPD78(F)094X”, Document Nr. U12097EE1V0PL00

4? NEC: „Preliminary Product Information µPD78(F)094X“ Document Nr. U12375EE1V0PM00

5? Jens Eltze: “Double CAN Controller as Bridge for Different CAN Networks”, Sponsors Session in the 4th CAN Conference 1997