iCC 1994

1st international CAN Conference

in Mainz (Germany)

Sponsored by

Allen Bradley
National Semiconductor
Philips 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

Craig Szydlowski, Intel Corporation

Tradeoffs Between Stand-alone and Integrated CAN Peripherals

The CAN Protocol[1] is currently implemented as on-chip peripherals integrated on microcontrollers and as stand-alone CAN chips. On-chip peripherals are available on several microcontroller architectures, including the MCS_ 51 and the MCS_ 96 microcontroller families. Likewise, there exists a variety of production-level stand-alone CAN chips such as the Philips PCA82C200 and the Intel 82527.

The decision to use an integrated CAN peripheral or a stand-alone CAN chip should consider the tradeoffs between both alternatives. These tradeoffs include implementation cost, design flexibility, level of CPU burden and system reliability. This paper discusses these tradeoffs from both qualitative and quantitative perspectives. The goal of this paper is to identify the key issues that differentiate these two alternatives for various design and production goals.

IMPLEMENTATION COST

The cost to implement a CAN peripheral in a system module may be divided into development and manufacturing costs. The development cost includes hardware and software engineering and design verification/qualification. The manufacturing costs include part procurement, product assembly and testing.

DEVELOPMENT COST - Figure 1 shows two hardware systems implementing CAN. System A requires three chips: a microcontroller or CPU, a stand-alone CAN chip and a CAN bus driver. The interface between the CPU and the CAN device is an address/data bus or a serial link such as the SPI protocol. A chip select signal is needed if other nodes

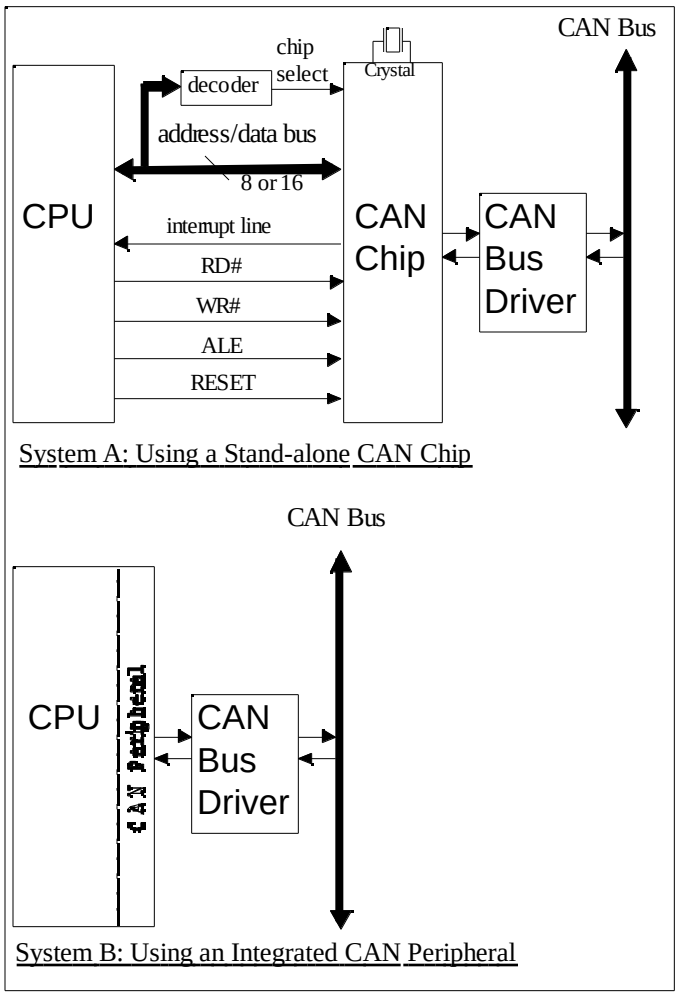

Figure 1: Hardware Implementations

are interfaced to the CPU's address/data bus. The CAN chip is driven by a low-tolerance input clock supplied by a crystal oscillator or the CPU's clockout. System A uses an interrupt line from the stand-alone CAN chip to the CPU to signal the reception of a message or the occurrence other CAN events.

[1]The Controller Area Network (CAN) protocol, developed by Robert Bosch GmbH, offers a comprehensive solution to managing communication between multiple CPUs. The CAN protocol is a serial communications specification for a multiplexed network to support control-oriented data. CAN implementations are prominent in automotive, agricultural, and industrial control applications.

System B implements a CPU with an on-chip CAN peripheral which clearly simplifies hardware design. In addition, system B often uses less printed circuit (PC) board area and generates less board noise by eliminating the PC board traces used to interface the CPU and the CAN chip.

The software engineering cost is nearly the same for integrated or stand-alone CAN peripherals. In both cases, software must be developed for the CPU to read messages following reception and to write messages for transmission.

MANUFACTURING COST - The manufacturing cost for a system with an integrated CAN peripheral is lower than a system with a stand-alone CAN chip. The integrated CAN peripheral has many advantages over the stand-alone CAN chip since one less chip is required, Figure 1. First, production expense is less for a system with an integrated CAN peripheral since there are fewer items to order and to assemble. Second, systems using integrated CAN peripherals require less PC board area than systems with stand-alone CAN chips which reduces system form factor. Third, module testing and repair are easier with the integrated CAN peripheral because there are fewer components to verify.

The semiconductor manufacturing cost of CPUs with on-chip CAN can be less than the cost to produce separate CPU and CAN chips. The silicon area for an integrated peripheral is less than a stand-alone chip since the CPU interface circuitry can be optimized for its host-CPU and no additional area is needed to accommodate packaging (bond pads and Electro-Static Discharge (ESD) circuits). The integrated CAN peripheral has a significant die size savings since it is about half the silicon area of its stand-alone chip counterpart. A CPU with on-chip CAN also benefits from a packaging savings by requiring one less package and fewer testing steps compared to the two chip alternative.

In high volumes, semiconductor manufacturers can offer CPUs with integrated CAN peripherals for a lower price than separate CPUs and CAN chips. Today, however, there are few high-volume applications implementing CPUs with on-chip CAN. Consequently, more generic and higher-volume CPUs

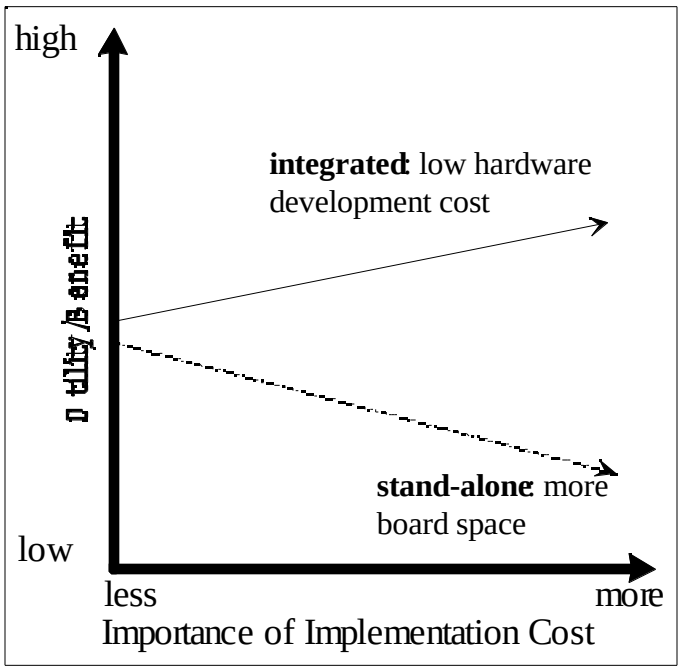

Figure 2: Importance of Implementation Cost

and stand-alone CAN devices enjoy production-economies of scale and possibly provide lower total system cost. As volumes for CPUs with integrated CAN increase, the manufacturing advantages through integration discussed previously will enable lower CAN node costs in the future.

Figure 2 shows a lower implementation cost for an integrated CAN peripheral. This trend considers the lower hardware development cost and the possible lower chip cost compared to the two chip alternative.

DESIGN FLEXIBILITY

Design flexibility allows engineers to upgrade their systems or to develop a similar system by applying existing hardware/software design experience. In either case, the flexibility to make changes often comes from exchanging the CPU to satisfy new requirements.

Although the software development is about the same for both stand-alone and integrated CAN implementations, software reusability may differ. Stand-alone CAN chips are designed to interface to different CPUs allowing the software developed for one system to be reused in another system, even if the CPU is different. Software developed for the integrated CAN peripheral of one CPU may not apply to a second CPU with on-chip CAN, especially if the CPUs are supplied by different vendors. This concept is shown in Figure 3 where the benefit from a stand-alone CAN chip configuration is more favorable when multiple designs are needed since the software and hardware development is reusable.

Therefore, exchanging the CPU to upgrade or to develop a new system may require some hardware modifications, but it is likely the same CAN chip may be used. By using the same CAN chip, the existing high-level language software may be reused, although the actual instructions differ among CPU architectures. The CAN chip has little impact on CPU interchangeability.

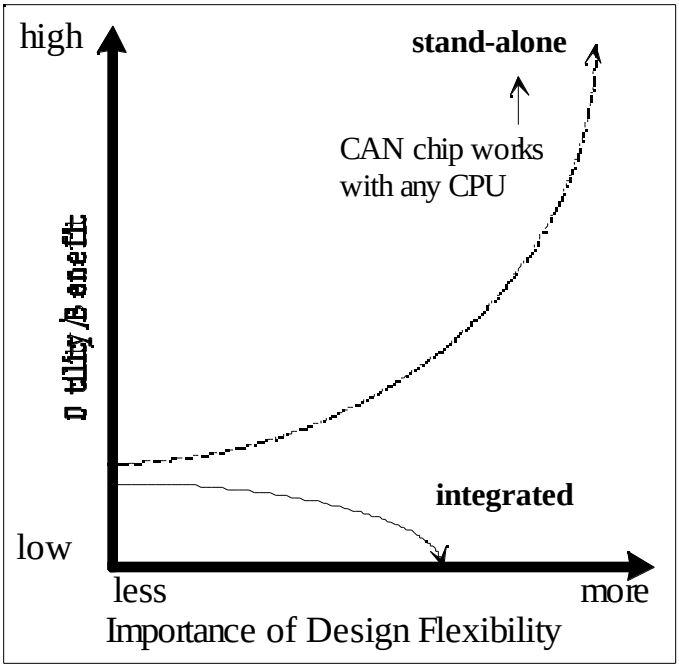

Figure 3: Importance of Design Flexibility

However an upgrade to a system that uses a CPU with on-chip CAN is significantly impacted by the CAN peripheral. The replacement CPU must also have on-chip CAN, otherwise a stand-alone CAN chip must be added to the system. Semiconductor makers are beginning to offer families of products with on-chip CAN providing users with CPU upgrade paths. Since the features of on-chip CAN peripherals are different among vendors, changing CPU architectures may require modifications to CAN software.

Some module makers must support a mix of networking protocols such as J1850, VAN and CAN. This flexibility is well supported by stand-alone protocol chips, through which only protocol chips vary among system module designs. In contrast, it is difficult to find a family of CPUs whose only difference is whether they have on-chip J1850, VAN or CAN.

LEVEL OF CPU BURDEN

Dedicated communications peripherals have been developed to support protocols such as CAN, VAN and J1850 to reduce the CPU burden to service a high-speed serial link. Even today's highest performance CPUs have difficulty supporting multiplexed communications above 100K bits per second using standard high-speed I/O pins instead of special peripherals. The level of CPU burden to maintain on-chip CAN is usually less than a stand-alone CAN chip since the CPU has faster and more efficient access to CAN registers. As a result, the CPU spends less time interacting with the on-chip CAN peripheral.

Figure 4 outlines the communications tasks at each CAN node with respect to the protocol, messaging and system/error response. The protocol tasks involve transmitting and receiving bits according to arbitration rules defined by the CAN protocol. Another protocol task is to calculate a 15-bit CRC code (cyclical redundancy error code) which is transmitted with each message and is verified at each CAN node. CAN peripherals complete all protocol tasks without CPU intervention.

With respect to the level of CPU burden, the messaging tasks must be serviced by the CPU. Messaging tasks require the CPU to write data to be transmitted, to read received data and to manage status/control registers in the CAN peripheral. Since the CPU "sees" the CAN peripheral as a smart RAM, messaging tasks are basically CPU read/write operations. A CPU with on-chip CAN will read/write to register locations using an internal bus. For a CPU interfaced to a stand-alone CAN chip, these read/write operations typically use the external address/data bus or a serial link. In addition to these read/write operations, the CPU may be required to manipulate message identifier bits and data fields as required by the messaging scheme. For example, the data byte may actually
contain multiple parameters such as engine air flow and engine temperature. In this case, the host-CPU must execute shift bit and masking operations to prepare and interpret complicated byte configurations. The CPU burden required to manipulate message identifiers and data bytes is a function of the messaging scheme, and this burden is the same for on-chip and stand-alone CAN peripherals. The CPU burden differs for on-chip and stand-alone CAN only because the access time of CAN registers is different.

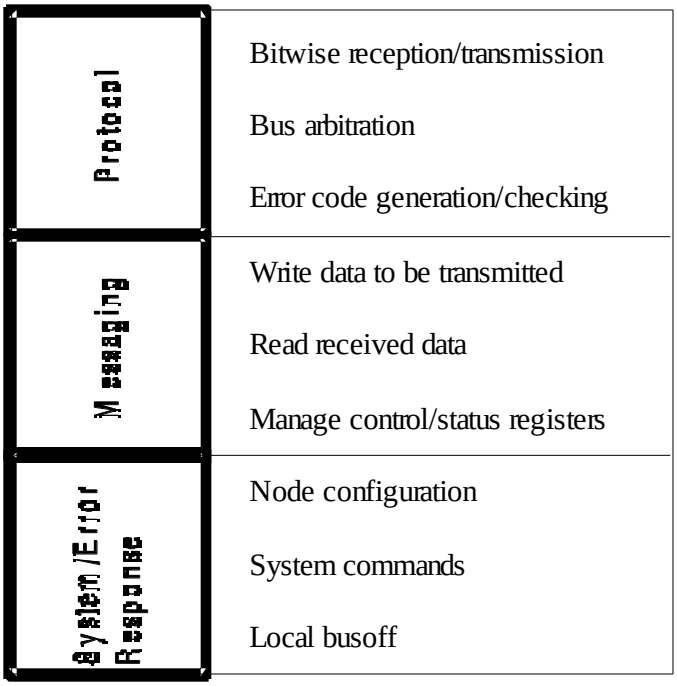

Figure 4: CAN Node Communications Tasks

System/error response is a general category for infrequent tasks initiated by the system or by an unusual number of bus errors. For example, the system may require nodes to dynamically allocate message identifiers during system initialization when similar nodes such as lamps are on the bus. The CPU also executes error recovery routines when the CAN peripheral is in "busoff state". Recovery from "bus-off" requires a hardware or software reset of the CAN peripheral.

The CPU-burden to communicate with the CAN peripheral is dependent on a few factors. The most critical factor is the amount of time required to read/write to the CAN peripheral. In the case of an on-chip CAN peripheral, the CAN registers are addressed using the internal address/data bus designed for high-speed access. In the case of a stand-alone CAN chip, the CPU uses its external address/data bus or a serial communications link. Figure 5 shows the level of CPU burden to receive CAN messages for three CAN bus transmission rates.[2] This analysis compared the CPU burden of an Intel 82527 stand-alone CAN chip to an Intel 87C196CA 16-bit microcontroller with on-chip CAN. The 82527 supports both 8-bit and 16-bit interfaces. The 87C196CA addresses the on-chip CAN as either scratch RAM or high-speed register RAM.

Stand-alone CAN Chip	CPU Burden 250KBS	500KBS	1MBS
8-bit A/D Bus	**5.5%**	**11.0%**	**21.9%**
16-bit A/D Bus	**4.2%**	**8.4%**	**16.7%**
Integrat. CAN Peripheral	**CPU Burden 250KBS**	**500KBS**	**1MBS**
Scratch RAM	**3.6%**	**7.2%**	**14.4%**
Register RAM	**2.0%**	**4.0%**	**8.0%**

Figure 5: Level of CPU Burden For Various CAN Nodes

The level of CPU burden to receive messages for an integrated CAN peripheral ranges from 2.0% to 8.0% whereas a stand-alone CAN chip requires 4.2% to 16.7%. [3]The CPU burden for an on-chip CAN is approximately one-half the burden of a stand-alone CAN chip. The 87C196CA accesses 16-bits word in 400nS clocked at 20MHz whereas the 82527 16-bit word read access time is 1300 nS.

In some cases, message filtering techniques can be employed to reduce the number of messages a node is required to receive in a given period of time. This will reduce CPU-burden by limiting the number of interrupts to the CPU by "screening" unnecessary messages. During system development, it is necessary to model worst case message reception and the resulting CPU-burden so the system functions properly when the CAN bus is heavily loaded.

The acceptable level of CPU-burden to service a CAN peripheral is application dependent. In some cases, a system may only be required to receive a small range of messages.

[2]The level of CPU burden assumes a maximum bus loading, 100%, an average message length of 100 bits (8 data bytes) and the minimum number of CAN accesses and CPU operations necessary to receive a message (management of the interrupt pointer and control registers).

[3]This comparison assumes 16-bit word operations, three wait- states access of the 82527 and 4 _S for interrupt overhead.

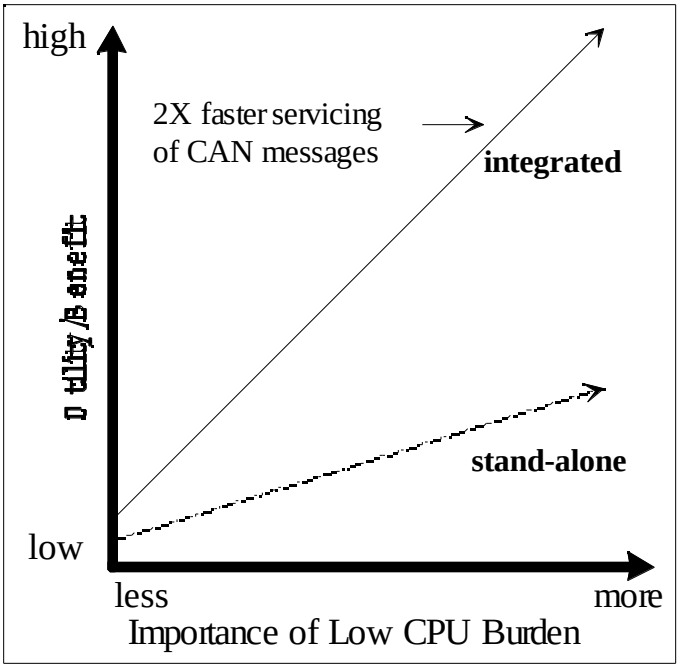

Figure 6: Importance of Low CPU Burden

An engine controller, on the other hand, may need to receive and transmit a large number of messages with high repetition rates.

Figure 6 indicates the lower CPU burden of on-chip CAN may provide a significant system advantage.

SYSTEM RELIABILITY

The reliability of an electronic system can be analyzed by considering the impact of the printed circuit (PC) board and the semiconductor components. A CAN node using an integrated CAN peripheral has a reliability advantage over a system using a stand-alone CAN chip because of its smaller form factor. As Figure 1 shows, the PC board implementing an integrated CAN peripheral does not have an interface between the CPU and the CAN peripheral and therefore requires fewer traces and plate through holes. An integrated CAN peripheral may provide improved chip-reliability over a stand-alone implementation since the integrated peripheral is burned-in and tested together with the CPU.

PC board reliability is a function of thermal stress. In particular, high use-temperature and temperature cycling aggravate the thermal expansion mismatch between the materials of the PC board by stressing the metal conductors and epoxy insulation. This additional stress increases the probability of cracks in the conductors leading to board opens. Temperature also accelerates insulator breakdown leading to shorts between board traces. Temperature cycling increases material fatigue caused by repeated expansion and contraction cycles.

"Plate through holes are Achilles' heel of the PC board, since they consist of thermally incompatible materials and are the basis of multi-layer interconnection. The second major cause of PC board failure is the loss of electrical insulation, termed insulation-resistance (IR) failure."[4] It is clear that PC board reliability decreases with additional traces, solder joints and plate through holes. Therefore, integrated CAN peripherals provide PC board reliability advantages.

The second aspect of CAN node reliability is the reliability of the individual semiconductor chips. CAN nodes implementing integrated CAN peripherals will typically have higher reliability than those implementing stand-alone CAN chips. First, the integrated CAN peripheral requires fewer circuits to interface to the CPU. The CAN peripheral is connected to an internal CPU bus eliminating the need for complicated interface circuitry and drivers. In contrast, the stand-alone CAN chip requires the integration of additional address/data bus circuitry, pin logic such as input/output circuits, Electro-Static Discharge (ESD) protection devices and mode select pins. As a result, an integrated CAN peripheral requires less interface circuitry and requires many fewer pins that could be susceptible to soldering issues and ESD exposure. The interface between the CPU and a stand-alone CAN chip requires as many as 21 traces or 42 solder joints for pin connections. In addition, the stand-alone CAN chip may require a separate crystal oscillator which impacts system reliability as well.

Chip reliability is improved by thorough testing and burn-in. An integrated CAN peripheral is tested and burned-in with the CPU, exactly as the chip will be used in a system. The integrated CAN chip is from the same silicon processing as the CPU, so manufacturing variability is assessed during testing.

Although a stand-alone CAN chip is certainly tested and burned-in, the CAN product engineer must guarantee operation with a variety of CPUs which is a complex product engineering task. The stand-alone CAN may be interfaced to a CPU from a different manufacturing process and possibly from different vendors. Fortunately, most hardware designers consider worst case chip specifications to ensure designs operate across chip manufacturing variations.

A CPU with an integrated CAN peripheral will have a specified reliability based on the results of the product qualification and the historical trends of the

[4] Clyde F. Coombs, Jr., Printed Circuits Handbook, Third Edition (New York: McGraw-Hill Book Company, 1988), p. 30.1.

manufacturing process. A typical defect per million (DPM) specification is perhaps 50-200 DPM. Measuring a lower DPM that is statistically significant requires a very large sample size and several additional months of evaluation.

A two-chip solution such as a CPU and a stand-alone CAN chip may actually have twice the DPM specification since the reliability of both chips must be summed. Even though the actual chip reliability of the stand-alone CAN system may be equal to the integrated CAN system, this is difficult to verify using reliability specifications derived on a chip-by-chip basis. Therefore, a CAN system with an integrated CAN peripheral may have better specified reliability than a system with a stand-alone CAN chip because of statistical limitations of comparing the reliability of single versus two-chip solutions.

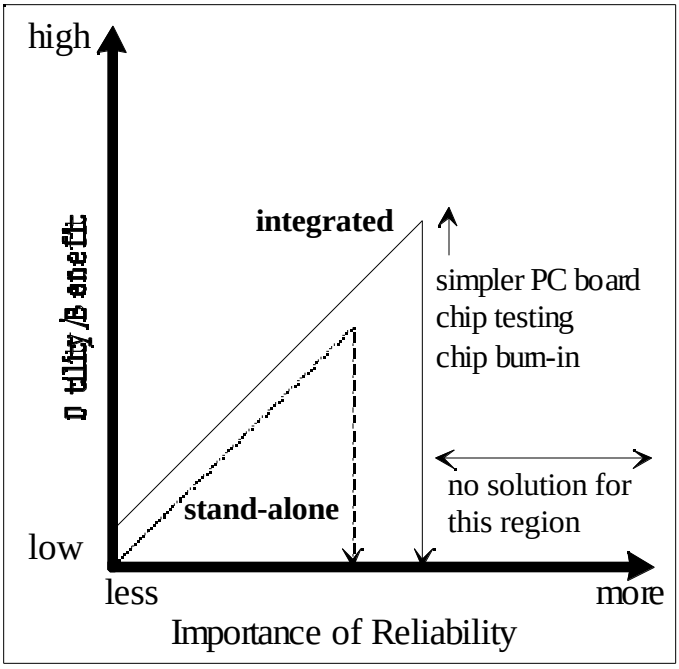

Figure 7: Importance of Reliability

Figure 7 shows the benefit of the additional reliability of a CPU with integrated CAN. The utility of both solutions falls when a very high level of reliability is required; this conveys the concept that an implementation which does not satisfy reliability targets is an unacceptable solution.

OVERALL ANALYSIS

The tradeoffs between using a stand-alone CAN chip and an integrated CAN peripheral may be viewed as the value of design flexibility versus implementation cost, level of CPU burden and system reliability. Yet today, many stand-alone CAN chips are shipping in high volumes, and the combined price of a CPU and a stand-alone CAN chip is very competitive. However, as CAN gains more market acceptance, on-chip CAN will be the solution of choice.

Figure 8 shows besides design flexibility, a CPU with on-chip CAN has many advantages compared to stand-alone CAN chips.

FOUR SCENARIOS

Four scenarios are presented to clarify design and production environments that weigh the tradeoffs between stand-alone and integrated CAN differently.

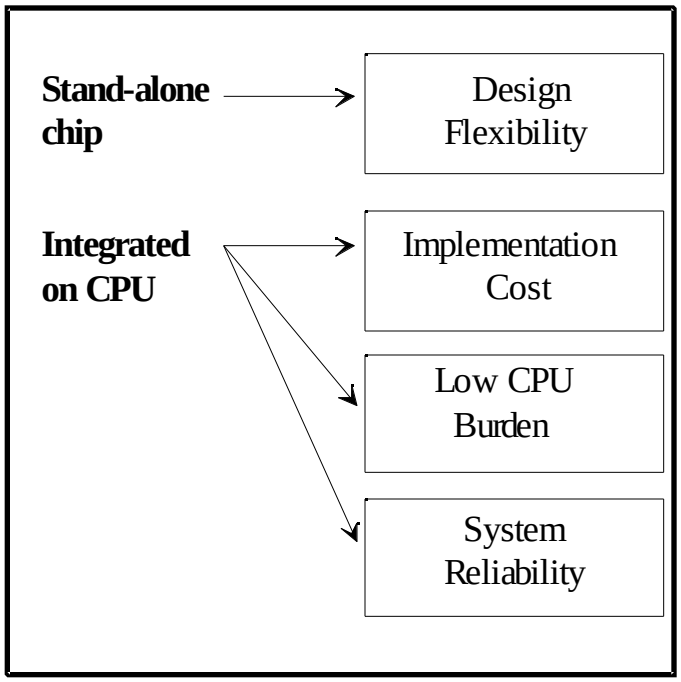

Figure 8: Tradeoffs Between Integrated & Stand-alone CAN

Scenario 1:

An engine control unit (ECU) manufacturer supplies a diverse set of car makers for 4, 6 and 8 cylinder engines. To meet various performance goals and control requirements, several CPUs from different architectures and vendors are used. Future emissions regulations will require system upgrading by implementing CPUs with more RAM and ROM needed for new diagnostic algorithms.

A stand-alone CAN chip provides this ECU maker the flexibility to change CPUs to meet cost targets while satisfying a range of performance targets. By using the same CAN chip for all engine designs, the CAN

hardware and software development is applied to all of the ECU designs
which minimizes product line development costs. The reliability of the ECU is critical to ensure automotive safety, and in almost all cases, a stand-alone CAN design satisfies system targets.

Scenario 2:

Anti-lock braking systems (ABS) become standard equipment on most passenger cars and shipment volumes soar while module prices to car manufacturers drop. Most ABS designs are mature and developers work to strip out costs from their modules. PC board size must be reduced, assembly costs cut, and procurement costs minimized to ensure reliable high-volume manufacturing. Reliability is a key requirement because braking is a primary aspect of vehicle safety.

An integrated CAN peripheral on a high volume CPU provides the lowest CAN node cost. The CPU with on-chip CAN is lower cost than the two-chip alternative, PC board size is reduced, chip procurement cost is minimized and the added performance allows a less powerful CPU to run the system. The integrated CAN peripheral provides increased reliability as well. Since design flexibility is not needed for a mature high-volume product, a stand-alone CAN does not offer any advantages over an integrated solution.

Scenario 3:

A robot is designed in the event a nuclear accident occurs and unmanned cleanup is required. This robot is controlled by a dozen of networked microcontrollers providing a range of motion. Reliability is a critical concern given the potentially grave nature of the situation. The high temperatures associated with the nuclear waste increases the possibility of electronic malfunction, so reliability issues are carefully addressed. Few robots must be manufactured, so system cost is not a primary concern.

A high-performance CPU with integrated CAN is best to meet the reliability needs of the robot. The compact size of this CAN node contributes many reliability advantages.

Scenario 4

A maker of in-vehicle climate control is informed that a new car platform will control his module through commands transmitted on a CAN network. The current systems use one of three 8-bit CPUs from a common architecture, but with various memory configurations. About one-half million lines of software have been written to support thirty different vehicle types. Even though the 8-bit CPUs have satisfied all climate control applications to date, there is little CPU headroom to service a CAN node.

A new 8-bit CPU with on-chip and more program memory is the easiest design change. An integrated CAN peripheral will burden the CPU less than a stand-alone CAN chip because the on-chip CAN registers are accessed faster. With the added CAN functionality, the climate control maker should request additional program memory to store the CAN subroutine.

CONCLUSION

Today, a large variety of CAN peripherals are available as stand-alone chips and integrated on CPUs. Integrated CAN peripherals offer a number of advantages over stand-alone CAN chips such as lower implementation cost, lower CPU burden, and higher system reliability. The stand-alone CAN chip offers one compelling advantage over the integrated CAN peripheral: design flexibility. Until CAN becomes a standard peripheral on all new microcontrollers, stand-alone CAN chips will serve designers who use a variety of CPUs and regularly upgrade designs.