Expanded single-chip principles in practical application

by RANDY M. DUMSE
Rockwell International

ABSTRACT

For the past two decades the semiconductor industry has been in a headlong rush to pack more and more features on a single piece of silicon. The creation of the microprocessor as a single LSI device naturally gave inspiration for further advances. The microcomputer on a chip followed quickly and was again a technological stepping point rather than a final goal. New generations and process development variations made possible larger, faster, and more powerful systems on a chip. There is, however, a limit on the amount of CPU, ROM, RAM, and special-purpose devices that can be placed on a single, easily manufactured silicon die with current technology. In order to give the cost-reducing features of a one-chip computer with the flexibility of a multichip set, the expanded single-chip computer was developed. This paper will explain the theory behind that development, and then explore its application in a specific example.
For the past two decades, the semiconductor industry has been in a headlong rush to pack more and more features on a single piece of silicon. The creation of the microprocessor as a single LSI device naturally gave inspiration for further advances. The microcomputer on a chip followed quickly and was again a technological stepping point rather than a final goal. New generations and process development variations gave larger, faster, and more powerful systems on a chip.

Along the way, many applications already designed in multichip systems were redesigned using advanced generation single chip microprocessors to take advantage of substantial systems cost reduction. Other designs which were not cost-effective previously in multichip versions were plausibly marketable, with one-chip computers providing substantial reductions. Of course, there are some applications where only a one chipper will suffice due to size, weight requirements, etc. The solutions in these areas using single-chip microcomputers have grown in number and complexity as the more sophisticated parts have become available.

Still, between the realm of what has been and what could be, current applications of microcomputers to the world in which we live have barely scratched the surface. The future will bring new and exciting designs. These designs will make possible consumer products that will challenge the imagination of man while easing his burdens.

Limiting factors

A closer look at the reason we are no further along in that endeavor will show three major facets that regulate advances. The first is time. Time moderates progress in several ways. Most obviously, as new microelectronic devices are perfected by the semiconductor manufacturers, there will be an appreciable delay before ideas on their use come into hand. Investors, engineers, and entrepreneurs will come together within the business world and move their dreams from design to production and distribution. The span between concept and product is time. It is less apparent, however, that time not only modulates the activity of these people but also their numbers. Educational systems cannot keep pace with the production of industry. There are more positions needing design engineers than there are design engineers. It is important to remember the reason for this phenomenon. Industry has found more ways of condensing features and functions on a piece of silicon than educational facilities have found to cram equal amounts of understanding of the use of these features into a single human head.

The second factor controlling progress is the level of advancement of the currently available microcomputer hardware. Along the scale of what can be implemented (at least some form of computerized electronic system) and what cannot, single-chip computer systems fall far short of center. The reasons are obvious. There are only so many CPU, ROM, RAM and special-purpose devices that can be placed on a single, easily manufactured, silicon die with current technology. Certainly, technology will increase production capabilities, but it is probably unreasonable to expect a single-chip microcomputer with over a kilobyte of RAM in the next two years, for example.

The last controlling factor to be mentioned is, of course, cost. The principle of anything which costs nothing and does everything will make the inventor a millionaire applies here. Overall system cost has limited many ideas from becoming realities. Certainly, if electronic calculators were still being done in costly multiple LSI sets, there would be several orders of magnitude fewer of them in the world today. Many applications which will become commonplace are unknown today because of cost.

Stretching the limits

Although time is an uncontrollable factor, system sophistication and cost factors are not. A closer examination of both is warranted. First, it should be pointed out that, to date, these two items have been counter points. System sophistication could not be improved substantially while independently reducing cost (at least while remaining at a given technological level).

Sophistication is generally improved by the addition of features. These may include new instruction sets or even revised architectures in the CPUs, more RAM and/or ROM, more input/output lines, addition or expansion of special purpose devices such as counter/timers, edge sensitive lines, latches, PLA's and the like. Almost all of these added features require their own portion of silicon. The more silicon per chip, the greater the likelihood of a small imperfection ruining that entire chip, resulting in lower numbers of good parts (from both less die per wafer and a higher degree of failure) and increased cost per chip.

Costs are generally held down with several techniques. The cost of the single-chip computer itself may be insignificant compared to that of the overall system. The amount of support hardware surrounding the microcomputer will to some degree be determined by the complexity of the applications. It is not always as obvious that the microcomputer itself may determine the cost and complexity of the support devices.
Internalizing more functions in implemented hardware or programmed software will reduce production costs. Of course, the programming required by such an approach will probably increase the engineering effort, but this added cost can be amortized over the production run.

Ideally it would seem every possible combination of ROM, RAM, I/O and special purpose devices like A to D converters, etc., should be included on a one-chip if maximum cost savings are to be realized. The assumption is based on a false economy, since such a device would be too large to manufacture with current technology or unique enough to have only one possible user. Remember, the main reason for using a microcomputer over discrete logic is the cost savings found in doing a custom programming of an existing part over a custom layout of a new logic design. It is the case then that an optimization between the device manufacturer and user must occur if both are to realize maximum profit (from reduced cost). The manufacturer should offer only a few options of microcomputers, the range of which combines the most often desired features in the best proportion for most users. This will ensure high volumes and low prices for the parts. There will, of course, be applications where these high-volume-oriented designs simply do not have the resources to handle the job. Now the cost balance between a custom-chip or a multiple-chip set must be made.

EXPANDED SINGLE-CHIP PRINCIPLES

The above discussion highlights the need for a compromise between single-chip and multiple-chip sets. A scheme is needed to give the cost-reducing features of a consumer one-chip computer with the flexibility of a multichip set. If an external bus structure were available on a single-chip computer, the problem would be solved. When the microcomputer did not have sufficient internal ROM, RAM, I/O and/or special function devices, they could be added externally. Cost would be held down by virtue of the fact that only the extra devices needed would be added externally, reducing chip counts.

This is exactly the principle of expanded single-chip computers. Designs already exist that not only incorporate a good deal of computing power on a chip with the support devices for most common applications included internally, but also allow flexible expansion externally. Two such microcomputers are the Rockwell R6500/1-11 and the R6500/1-41. A detailed look at these devices is in order.

The Rockwell R6500/1-11

The R6500/1-11 (called the R-11 hereafter for simplicity), is one of the most advanced multifeature one-chip microcomputers available commercially. Based on an enhanced version of the R6502, the part has an extremely powerful 8-bit CPU with four new instruction set groups added. These groups are Set Memory Bit (SMB), Reset Memory Bit (RMB), Branch on Bit Set (BBS), and Branch on Bit Reset (BBR). These new instructions, coupled with the parts, high level of throughput (one µs minimum instruction cycle time), give an I/O intensive and very powerful general purpose microcomputer. A generous portion of 3K bytes of ROM is designed into the chip. Also, 192 bytes of RAM are provided. In the 64 pin QUIP up to seven I/O ports are available, each with 8 individual lines for a total of 56 lines. Four of these lines can act as edge sensitive inputs. A complete, double buffered, full duplex, advanced feature serial channel is incorporated in the part. It will operate either synchronously or asynchronously. The inclusion of two 16-bit timers, one with a 16-bit latch and one with two 16-bit latches with multiple modes, gives the device many real-time signal processing and generation capabilities. This brief listing does not mention all the features of the R-11 but does point out that the designers included as much capability on a single chip as is feasible. To accommodate applications where these features are not sufficient to meet the product designers’ needs, they also included two external bus modes that are program selectable so that external parts could augment a one-chip microcomputer.

The first of these modes, the Abbreviated Mode, provides an external data bus and six address lines, as well as the control signals required to affect data transfers. This mode supports 64 external locations and is most suitable for the addition of memory mapped I/O or special function devices. The second mode, the Multiplexed Mode, provides fourteen addressing lines, eight of which must be latched, as they time-share with the data bus. This mode gives a 16K contiguous memory map external to the part. Any type of device such as ROM, RAM or special function I/O device could be accommodated singly or in combination.

The Rockwell R6500/1-41

The R6500/1-41 (called the R-41 hereafter for simplicity) is an interesting device which can be characterized as an Intelligent Peripheral Controller (IPC). Designed to reside on a host processor’s memory or input/output busses, this device can be programmed to control a given set of preassigned real world tasks. Although it contains an enhanced 6502 central processor of its own, it would appear to the host as a special purpose input/output or control device, as would any other LSI controller device such as a floppy disk or CRT controller. Based on the control and data words written into the R-41, it could execute commands and sequences programmed in its 1.5K internal ROM. Also available are 64 bytes of RAM. Besides the three state port on the host bus, the R-41 can host up to 6 input/output ports or 48 individual I/O lines in the 64 pin QUIP version. Two of these lines have edge detect circuitry. Much like the previous generation R6500/1 single-chip computer, the R-41 also hosts a multifunction, multi-mode 16-bit counter/timer with full 16-bit latches. Like the R-11, the R-41 has two external bus modes of its own and can support other LSI controllers or memory in its own memory map. Although the R-41’s external bus modes have the same names as those of the R-11, there is a slight variance in function between the parts. The Abbreviated Mode of the R-41 has four address lines and two control signals. This provides for 16 contiguous external memory address. The Multiplexed Bus Mode provides an additional eight data lines time multiplexed
on with the data bus. This provides a full 4K of external memory map for RAM, ROM or devices.

EXPANDED SINGLE-CHIP APPLICATIONS

To highlight these expanded single-chip computer principles, a specific example of possible application will now be explored. Consider the current market state of electronic typewriters. Most are still largely mechanical with servo enhancement of the operators keystrokes. A great deal of mechanical complexity could be replaced by microprocessor logic and a cost savings realized. In all likelihood, improved features could be added with little additional effort. A specification will be formulated in the following paragraphs to make good use of the R6500/1-11 and R6500/1-41 features in this application.

Application specifics

The actual printer mechanism to be considered will be a daisy wheel type. The most basic of features will require scanning of the keyboard and control of the printer servo mechanical devices. The wheel motor and position timing, hammer timing and control, carriage positioning (left and right), and platen control (paper advance or positioning) are included. A typing speed of 10 characters per second more than covers the speed at which an above average typist could enter keystrokes. This would be the equivalent of about 120 words per minute, so this will be the basis for all timing specifications.

The print wheel timing and positioning could be accomplished in a number of different ways. Almost all of these combinations, however, fall into two categories, i.e., either a stepper motor to give a character change per step or a D.C. servo. Both would require a start point reference input.

Hammer control would be used once the daisy wheel was in position for the impact. Since different size letters would print with a different tonal intensity if the impact was not calibrated, the force used to strike the letter must be modulated. This might require combinations of different coils or energizing a single coil with different width pulses for the different characters.

The common type spacings are pica or elite, which place a character every twelfth or tenth of an inch. The common denominator between the two type sizes is 120th of an inch. Assuming a stepper motor was used to position the carriage, it would make 10 to 12 steps to move between character positions (⅙ of an inch is also possible).

Even if subscript or superscript positioning were required, the mechanics of the paper feed could be fairly straightforward and done with a single stepper motor.

To meet requirements of printing 10 characters per second, all the functions of paper movement, carriage positioning, wheel positioning, and hammer impact would have to be accomplished within 100 milliseconds. Of course, other functions would be going on concurrently. The keyboard must be scanned every 20 milliseconds or so in order not to miss any key closures.

Design details

Beyond the most basic requirements, many other features are possible when microprocessing power is added to the system. A single-line display, correction and editing of a line prior to printing, page-at-a-time memory, interfaces to mass-storage devices, and even computer interfaces are possible at very little cost difference over the basic typewriters. These additional features will make good examples of the expanded microcomputer principles and will, therefore, be included in this example specification.

This design will include, therefore, a single-line, 80-character display unit. It will allow an entire line to be entered on the display before it is printed. It will also be memory expanded and will "remember" an entire document, up to four pages of typed material. Once the document is in this memory, the typist will be able to review and make any corrections needed prior to reprinting. An RS232 channel will also be included to allow communications with a host computer or RS232 compatible mass-storage devices. It will, therefore, be useful not only as a typewriter but also as a computer terminal, a data recorder, and a limited-application stand-alone word processor.

Now that the requirements are stated, the details of the implementation can be revealed. Although there are many possible combinations of R6500/1-11's and R6500/1-41's that could meet these needs, the design selected here represents only one. It is offered only as a reasonable example. A single R-11 would host the entire system (see Figure 1). Support devices, as needed, reside on this part's external bus. In this particular implementation, the multiplex bus mode will be

Figure 1—Typewriter block diagram

From the collection of the Computer History Museum (www.computerhistory.org)
selected on the R-11 to allow a full 16K bytes of external memory to be addressed. In that address range of the system host will be 8K bytes of RAM, one ROM chip, and two R-41's.

System description

The tasks assigned to the host include the scanning and processing of all keyboard and panel switches, management of the RS232 serial port, maintenance of the entered document in RAM, performing the word processing functions, and commanding the actions of the two R6500/1-41's. One of these R41's is assigned to control all the stepping motor functions. The other is dedicated to the display. The tasks are organized in this manner to reduce the impact of small changes in the mechanics and display units on the overall system. This will allow future models to use more elaborate features in these areas without requiring any modification of the host system. Only the R41 involved with that portion would need reprogramming. As such, a great cost savings in new development could be realized.

The keyboard is a matrix of 47 alphanumeric key caps arranged in standard QWERT format with typewriter placement of the shifted characters (as opposed to teletype). Ten additional keys are required, comprised of the BACK SPACE, LINE FEED, RETURN, DEL, ESC, TAB, CTRL, LOCK and two SHIFT keys. A numeric key pad area and several selection buttons for system control are not required, but desirable for typewriter operations. Some of these additional keys can be in the matrix while others should occupy individual input positions. The CTRL and SHIFT keys are examples of the latter, since they will be closed simultaneously with other keys in the matrix.

Although there is some controversy about the type of rollover processing that is really required in a keyboard operated by a high-speed typist, N-key rollover is still the most popular and the reigning standard. Some terminal manufacturers are beginning to turn away from the concept, using 2-key rollover or lockout instead. N-key rollover programming requirements are considerably more complex for little or questionable performance improvement. Still, because it is the highest standard, the N-key design will be used in the example.

The RS232 port will be very easily implemented by using the serial port of the R-11. All the features of this channel are programmable to meet almost all common applications (including parity as required). When the typewriter LOCAL switch is active, the RS232 port could be connected to a selected RS232 compatible mass-storage device. Many such devices are available, the most suitable for this application probably being data cassette types. On command from the keyboard, the document contained in memory could be stored on the data tape. If it were desirable to review or edit it, the saved document could be retrieved later from storage. If not in the local mode, the key functions and printing would be independent. Keys depressed would be passed from the typewriter to an external device. The external device returns would be printed as received. This is exactly the essence of a full duplex terminal. Instead of a video screen for display, however, the output would be letter quality print.

Since internal RAM is limited to 192 bytes, it is necessary to expand the RAM with external parts. The internal RAM will be used for the processor stack and system constants such as tab settings, margins, etc.; and system variables will be used for calculations, keymask patterns, and the N-key stack. In order to provide N-key rollover, two images of the keyboard must be maintained. The differences from one scan to the next represent the new key information. As each new key is depressed, it is added to the N-key stack. A keyboard matrix can be rather large, so the two images will be stored in external RAM. In order to enter a line at a time and also do editing functions, a current line buffer will also be maintained in external RAM. A page of typed information requires 2000 bytes of storage or less, so 8K bytes are necessary external to the R-11 host.

The 3K bytes of internal ROM in the R-11 will be sufficient for the management of all features and communications with the possible exception of the keyboard key cap assignments for the SHIFT and CTRL combination, if the pattern is non-standard, and perhaps some of the more complex text editing features that might be added. These features’ programs could be maintained in an external ROM. It is doubtful that anything larger than a 4K byte ROM would be needed even if the features included centering commands and formatting with pagination functions.

The RS232 port will be very easily implemented by using the serial port of the R-11. All the features of this channel are programmable to meet almost all common applications (including parity as required). When the typewriter LOCAL switch is active, the RS232 port could be connected to a selected RS232 compatible mass-storage device. Many such devices are available, the most suitable for this application probably being data cassette types. On command from the keyboard, the document contained in memory could be stored on the data tape. If it were desirable to review or edit it, the saved document could be retrieved later from storage. If not in the local mode, the key functions and printing would be independent. Keys depressed would be passed from the typewriter to an external device. The external device returns would be printed as received. This is exactly the essence of a full duplex terminal. Instead of a video screen for display, however, the output would be letter quality print.

Since internal RAM is limited to 192 bytes, it is necessary to expand the RAM with external parts. The internal RAM will be used for the processor stack and system constants such as tab settings, margins, etc.; and system variables will be used for calculations, keymask patterns, and the N-key stack. In order to provide N-key rollover, two images of the keyboard must be maintained. The differences from one scan to the next represent the new key information. As each new key is depressed, it is added to the N-key stack. A keyboard matrix can be rather large, so the two images will be stored in external RAM. In order to enter a line at a time and also do editing functions, a current line buffer will also be maintained in external RAM. A page of typed information requires 2000 bytes of storage or less, so 8K bytes are necessary external to the R-11 host.

The 3K bytes of internal ROM in the R-11 will be sufficient for the management of all features and communications with the possible exception of the keyboard key cap assignments for the SHIFT and CTRL combination, if the pattern is non-standard, and perhaps some of the more complex text editing features that might be added. These features’ programs could be maintained in an external ROM. It is doubtful that anything larger than a 4K byte ROM would be needed even if the features included centering commands and formatting with pagination functions.

Figure 2—Example keyboard layout
All the features of the host have been described. Now attention will be turned to the slave processors functions. The two Intelligent Peripheral Controllers (R-41’s) manage the output functions of the system. The first to be discussed controls the printer mechanism.

This R-41 would receive characters and commands through its data port from the host R-11’s processor bus. The distinction between commands and characters would be made by the previously written control registers port. In this manner, the host would send characters in a stream to the R-41. The R-41 will determine spacing, paper feed, and print wheel and hammer control (in short, all the functions necessary to put the character to the paper). If special carriage control were required (line feed, carriage return, back space, superscript positioning, etc.), the host would send the specific associated control word instead.

The functions of the display could probably be processed by either the host R-11 or the printer R-41, but in order to give a flexible system design for future expansion as previously described, the second R-41 will be used to control the display. Such a design would also be advantageous by virtue of the fact that no additional I/O or timing burden would be placed on the host or printer subsystems. After all, the simpler the modules, the quicker the system can be completed at a lower development cost.

The interface between the host R-11 and the display R-41 would be nearly identical to that of the printer slaye processor.

Buffered with transistors, some of the R-41’s output lines could drive patterns for a long vacuum fluorescent display tube. Other port lines could be demultiplexed to give the select for a particular character position. Beyond these parts, only a power supply for the required V.F. display voltage would be needed to complete this module.

SUMMARY

The specification and design details are now complete, at least to the scope of this paper. Only six LSI MOS chips would be required for this entire system (an R-11, two R-41’s, two 4K x 8 quasi static RAM and one ROM). The cost of the parts in OEM quantities for this application is under $50. The entire electronics assembly with display could probably be made for under $90, meaning it is reasonable to conceive of a high-quality typewriter, terminal text processor that could be marketed for under a $400 retail price tag.

The principle of using expanded single-chip computers to reduce costs is, therefore, proven. Development of a custom processor chip to handle all the features described could approach one million, if possible at all in current technology.

A multichip set approach would at least double the chip count and probably the cost of electronics, while offering no additional features. The application of expanded single-chip computers fits the needs of today’s market.