(From the magazine *Elektronik,* December 1982, converted to text with OCR and then translated from Danish assisted by the translator at deepl.com)

Microcomputers in consumer products

By Arne Rohde,
Bang & Olufsen A/S

The latest tape recorder from B&O uses microcomputer technology to achieve operating comfort and advanced features.

Introduction

Bang & Olufsen has been developing and marketing consumer products with built-in microcomputers for a number of years. The first product on the market was the Beocenter 7000, and a number of advanced audio and video products have since followed. Common to these products is that microcomputers are primarily used to simplify operation for the user, and to provide functions that would be unrealistic to incorporate using conventional electronics.

The Beocord 9000, shown in Figure 1, is an advanced cassette tape recorder that is in part an evolution of the Beocord 8000 and Beocord 8002 with a number of improvements and new features. As Beocord 8000 is microcomputer controlled, it was logical to try to use the same microcomputer in Beocord 9000, reusing as many routines as possible to reduce development time and thus cost. The intention here is to review some of the problems that can arise in such a project, and the means by which they can be overcome.

Single-chip microcomputers

It is important to clarify concepts and terminology as there is some confusion within the industry. A microprocessor usually consists of a single IC (integrated circuit) containing the equivalent of the computing unit in a traditional computer. As a stand-alone IC it is useless, requiring some form of program and data memory, input and output devices, and various other circuits to be of any use. When these components are added, a microcomputer is created, which is a small form factor computing machine. Some years ago it became clear that all functions could be built into a single IC, which then required only a power supply to be used as a computing machine. These ICs became known as single-chip microcomputers, and have since become very widely used for embedding in various products, such as tape recorders, gramophones, radios, instruments, toys, sewing machines, washing machines, etc.

A single-chip microcomputer, hereafter referred to as microcomputer only, typically consists of a 40-pin IC with a 4- or 8-bit arithmetic unit, a program memory of approximately 500 to 4000 bytes (also called ROM or Read Only Memory), 20 to 256 bytes of data memory (called RAM, or Read And Write Memory), a clock generator, and 15 to 35 I/O pins. An I/O (input/output) pin is a pin on the IC that can be directly controlled from the application to a high or low level, or the application can read whether the pin is supplied with a high or low level from the outside. Some microcomputers can be extended with external circuits that can contain program and/or data memory, or multiple I/O pins. Microcomputers are available at prices from about $1 to $20, depending on size, function and quantity.

The Beocord 8000 and 9000 use an Intel 8049 microcomputer that contains 128 bytes of data memory, 2048 bytes of program memory, and 27 I/O pins in a 40-pin IC. In the Beocord 8000 this is extended by an Intel 8355 chip containing 2048 bytes of program memory and 16 I/O pins in a 40-pin IC, and in the Beocord 9000 two of these chips are used. When connecting such a chip, 12 I/O pins are used on the 8049 which thus cannot be used as normal I/O pins. This results in 31 I/O pins with one external circuit, and 46 I/O pins with two external circuits (one pin is used to select one or the other external IC).

Single-chip microcomputers can be based on a 'general purpose' microprocessor, such as the Rockwell 6500, Motorola 6801 and 6805, Mostek 3870 and TI9940, or be specially designed such as the Intel 8049, Zilog Z8, and most 4-bit microcomputers.

Common to all computers is that they require a program to perform a useful function. For single-chip microcomputers, it is common to write programs in a language called assembler, which can be translated line by line directly into machine code.

Program Development

Program development for microcomputers is done on a development system that includes a machine code translator (or assembler) that translates program code into the machine code of the microcomputer. In addition, the equipment will usually include an 'emulator' which can emulate the final microcomputer and which can be controlled from the development system. This emulation should of course be as accurate as possible, but often there will be small, but perhaps significant, differences from the final microcomputer. When executed in the emulator, the program can be interrupted at predefined points if desired, and the contents of the data memory and program memory can be read and modified. In this way, the program can be tested before the order for the final ICs is placed. Several of the microcomputers are also available in special versions where the program memory consists of an erasable memory that can be programmed in a special device, and these can be used for pre-production or for small batch production.

Once the program has been tested, it is sent to the IC supplier who produces a special version of the IC with the program as a permanent component, after which program changes are both slow and very costly, as a new IC must be produced to correct the program. There will often be 12 to 20 weeks from delivery of the program until delivery of large quantities can commence.

The development process

In any development project there will be different alternatives to be assessed and selected. For example, in the development of a new tape recorder:
- Should it be controlled by a microcomputer and if so, what type?
- Should it have separate or combined input/output heads?
- What noise reduction systems should be used?
- Should it have one or more motors for tape transport?
- What operating functions should it have?

In the case of the Beocord 9000, part of the answer was given in advance. It had to be controlled by a microcomputer, preferably of the Intel 8049 type as this is used in the Beocord 8000, and the controls had to be almost the same as in the Beocord 8000. However, it would have numerous new features which would mean a significant extension of the original program. It was also given that there should be an automatic adaptation of the recording on different types and brands of tapes. This adaptation was to be simple and not very time-consuming.

The problem with the Intel 8049 was that it could only address 4096 bytes of program memory, and these were fully utilised in the Beocord 8000. This was solved by having the program parts of the two external 8355 circuits in the same address range, with the selection of each circuit done in the processor by setting a selection pin high or low. Unfortunately, this was not possible to emulate in the development system, and the development time was therefore extended. Each new version of the program had to be loaded into EPROM (Erasable programmable read only memory) versions of the Intel 8355, called 8755. This is a time-consuming process. The EPROM version of the 8049 did not exist at the time, so it had to be emulated with an 8039 and an 8755.

Automatic adjustment of the recording parameters was also something new, and it was therefore necessary to test different alternatives. The system had to cope with all tape types and makes, so a large number of tapes had to be measured to determine the ranges to be covered. This was done by creating a special version of the program that could print the measurement results directly on a matrix printer. The tests showed that there could be large differences between tapes of the same type and make, and that the variation from one end or side of the tape to the other could also be large. This variation could be used to determine the required measurement accuracy of the final system.

The project was started with a single read/write head, and the program for this version was almost complete when a suitable integrated unit with separate read and write heads was found that could meet our requirements. With a single head it was necessary to have two passes of the tape to be measured, one to record the various tones on the tape and one to play them back. A total of 5 parameters are measured, but the two of these, bias level in the left and right channels, need to be established and set before the other three can be measured. Therefore, the first two were recorded and played back with an intermediate rewind, and the bias level could then be set in both channels. Then the last three sets of tones could be recorded and played back, again with an intermediate rewind. In total, this process lasted approximately 45 seconds.

With an integrated head, a tone could be recorded, and immediately afterwards played back and measured. At normal tape speed, the distance from the recording head to the playback head corresponds to only about 80 milliseconds of playing time. Within these 80 milliseconds, the recording must be stopped to avoid crosstalk from the recording head into the playback head, and the tape must be played back to measure whether the parameter is correctly set. This measurement is made 3 times at 16 millisecond intervals. A number of experiments were necessary to determine whether this procedure could be performed with reasonable measurement accuracy and reproducibility. The playback procedure is shown schematically in Figure 2. The record head labelled R (for Record) is shown on the left and the tape runs from left to right, from R to P which is the playback head. The recording of a certain level is stable after about 15 milliseconds, and then continues for another 85 msec. After the end of the recording it waits for 15 msec before the recorded signal is played back and checked. The signal is checked up to 3 times and all 3 measurements must be accepted before the level in question can be accepted. If even one of the three measurements falls outside the desired limits, the recording of the next level is started.

The use of the new integrated head meant a reduction in the total run time from 45 seconds to approximately 10 seconds. A significant additional advantage was that it can be quickly determined whether the measurement is made on the lead-in tape, and the tape type can be determined even if it differs from the code holes in the back of the cassette. It was also possible to incorporate a dynamic alignment procedure that could be used in production and in service workshops.

As the Beocord 8000 had provided a lot of experience and requests for new or modified operating functions, it was natural to build these into the new tape recorder. It would be too lengthy to go into these functions here, but the result was that the program was almost completely rewritten, both to reduce the size as much as possible and to fit in the new functions.

Microcomputer environment

The many I/O pins are all utilized, and in addition, various ICs are used to further extend the control capabilities. The microcomputer and its immediate surroundings are shown in Figure 3. The wide arrows consist of two or more pins or wires, the thin lines denote a single I/O connection.

For setting the recording parameters, five 4-bit counters are used as D/A (digital to analog) converters, each of which can be controlled from the computer. For each step to be counted, the circuit must have a short clock pulse applied, and it is necessary in the program to keep track of the current step number. A reset pulse can be used to set all the circuits to a known value. An additional circuit of the same type is used to select the various tone generators to be engaged. A 3 to 8 bit expander is used to select which of the 6 counters will respond to the clock pulse.

The measurement of the recorded tone is made by a comparator that can set an I/O pin high or low. This measurement is dynamic, and it is the computer that determines whether or not to accept a particular measurement.

The computer receives commands from a keyboard, which are constantly read, even when the device is in stand-by. These keys are read in groups of 5, and debounce is necessary to avoid errors on the keys that have dual functions. The result of a command can be reflected on a four-digit display, on one of the four lamps, or on one of the five LEDs under the lid. These are all controlled by the computer, which must multiplex quickly between the four digits to give the impression that they are on constantly and simultaneously. The digits are switched at intervals of about 2 milliseconds.

Commands can also be received from a wire that may be connected to a Beomaster. The same wire is used to send a message from the Beocord to the Beomaster. This data is sent as an 8-bit code with a start bit, 6 data bits, and a stop bit. This line must also be constantly monitored, and codes destined for other devices on this so-called data link must be ignored.

The voltage to the motor and to the bridge with record/playback heads is controlled digitally, and the current consumption is measured to avoid an overload if the motor or bridge become blocked. This measurement is disabled for a few seconds after the bridge is retracted or extended, as well as after motor start-up and at end of tape stops. The winding speed of the motor can be adjusted to avoid tape stretching at end of tape stops.

Tape time is displayed in minutes and seconds, but this must be calculated from pulses received from a pulse generator measuring directly on the pick-up reel. This provides a signal change 12 times for each revolution, or at approximately 2 millisecond intervals during fast winding. There is also a pulse generator on the let-off reel that is used when calculating the amount of tape remaining on that reel. When the let-off reel contains less than about 5 minutes of tape, a warning is given during recording.

The tape type is initially determined by holes in the back of the cassette, and two switches are provided to measure this, which can be read by the computer. Another switch indicates whether a cassette is mounted.

The computer can remember recording parameters for four fixed tape types, in addition to the current measurement. In the event of a power failure, it is not desired that these measurements be lost, but the data memory in the computer has a high power consumption that cannot be handled by an ordinary battery. Therefore, an additional CMOS type memory is built in which has a very low power consumption in standby mode. This can be kept alive for about 8 years by a small lithium battery. The measured data are stored both in the internal data memory and in the external memory, and can thus be reloaded after a power failure or other disconnection.

It’s also possible to measure whether there is a signal on the tape being played. This is used during searching where playback is activated when a pause is found on the tape. During recording and playback, the various signal paths and generators must also be controlled in a specific sequence to ensure noiseless switching of signals.

The unit has a built-in clock controlled by the mains frequency, which over long periods is more accurate than an untuned crystal. As the device must be able to operate in areas with a 50 Hz network or with a 60 Hz network, the computer itself determines whether it is connected to one or the other. The clock can be used to start a recording or playback at a given time, or to switch the device off.

For service purposes, the unit can be set to a special mode where the dynamic recording parameters can be checked and adjusted with a screwdriver as the only tool. The five parameters can be set and controlled individually in both left and right channels.

All in all, there are thus many inputs to be read and outputs to be controlled as soon as a particular command is received.

Program Structure

The program is structured in three main modules, each of which is divided into several other modules. When power to the Beocord 9000 is switched on, the start-up routine is executed, which sets all outputs to a known position, sets the internal data memory to fixed values, loads all information from the external CMOS RAM, and puts the device in stand-by position, ready to receive a command.

The program is interrupted every 1.04 milliseconds by an interrupt from the on-board timer controlled by a crystal, which also determines the instruction times. This interrupt routine forms a large part of the program and carries out all the time-critical parts of the program. For technical reasons, this routine must be located in the 8049 IC, and it uses the majority of its memory space.

The display values are updated every 64 milliseconds, and the output routine performs this. Here it is decided what should be displayed, such as tape time, clock, timer start or stop, blink for error operation, etc.

Reading of the keyboard is done every 32 milliseconds and if a command is received, this is executed in one of the subroutines. There is a subroutine for each key function.

Interrupt Routine

The interrupt routine executes the time-critical parts of the program. The following functions are performed in this routine:

Program status is saved.

A new interrupt is requested in 1.04 msec.

Values for display and lamps are read out every second time.

Values on input pins are read and stored in internal memory.

The data link is read for a new code, or a requested code is sent on the data link.

The input from the mains frequency is read and the clock is updated.

The pulse generator from the take-up reel is read.

Tape time is calculated if the tape is time calibrated.

Read the pulse generator from the let-off reel.

Winding speed is reduced if a given speed is exceeded.

The reel rotation time is measured during tape time calibration.

The tape counter is compared to a desired counter value during search.

The keyboard is scanned every 32 milliseconds.

The counter circuits for recording calibration are updated.

Time measurement values are updated.

Next value is written to the external CMOS RAM if there is enough time left of the 1.04 msec.

It is decided whether the input routine (every 32 msec) or the output routine (every 64 msec) should be activated.

The program status is restored.

Many of the measurements made are initiated by either the input or output routines. During tape calibration, the calibration routine determines which circuits to send clock pulses to and how many to send. The interrupt routine performs the clocking, and notifies when it is completed. This is done partly to avoid an interrupt occurring while the clocking is being performed, and partly because a program part in one of the external ICs does not have direct access to the I/O pins located on the other IC since only one of these can be selected at any given time.

Calculation of the tape time is the largest subroutine of the interrupt routine. Once the thickness is known, the tape time could be calculated directly, but this requires multi-digit multiplications and a division. Only a few microcomputers contain direct instructions for multiplication and division, which otherwise must be done in loops by repeated addition and/or subtraction. An estimate of the computation time shows that it would require about 100 msec to convert pulses to time on an Intel 8049, and since the pulses come at about 2 msec intervals during winding, this is an unrealistic approach.

Instead, tables are stored in the program memory containing the time consumed for a range of counter values, for each of the tape thicknesses used (C60, C90 and C120). Between the stored values, the conversion curve is assumed to be linear and a simple interpolation can therefore be used to find the correct value. This table look-up and subsequent calculation can be done in about 300 microseconds, and is therefore performed for each pulse from the take-up reel.

It would go too far to go through all subroutines in detail. The above details are only listed to illustrate some of the issues that need to be addressed when developing such a program.

Development Time

The program itself consists of about 3500 instructions, all of which need to be tested. This can be a very lengthy and demanding process, quite apart from the programming time itself. For the design, programming, testing and documentation of programs in machine code, figures from 10 to 30 instructions are often estimated for each working day. This would entail a time of 120 to 350 working days, or 1/2 to 1.5 years. Development began in the spring of 1980 and was completed by Christmas 1980, but since much of the program was rewritten after changing the recording/playback head, this probably equates to about 40 to 50 instructions per day.

Conclusion

The development of embedded software for consumer products is particularly demanding in several areas. Programming is done almost exclusively in machine code (assembler) to save space and time (execution time, not development time). Efficient use of program memory is necessary to make the product price competitive and to incorporate as many useful functions as possible and desirable. Efficient use of time is necessary in many cases for the program to perform time-critical routines in the time available. The microcomputers used often have an instruction set intended for control tasks, and calculations can therefore be complicated to perform.

The use of microcomputers and processors in various products will increase in the coming years, both because of the cost and the functions that can be performed, and there will therefore be a growing need for staff with knowledge of both digital electronics and computer programming.

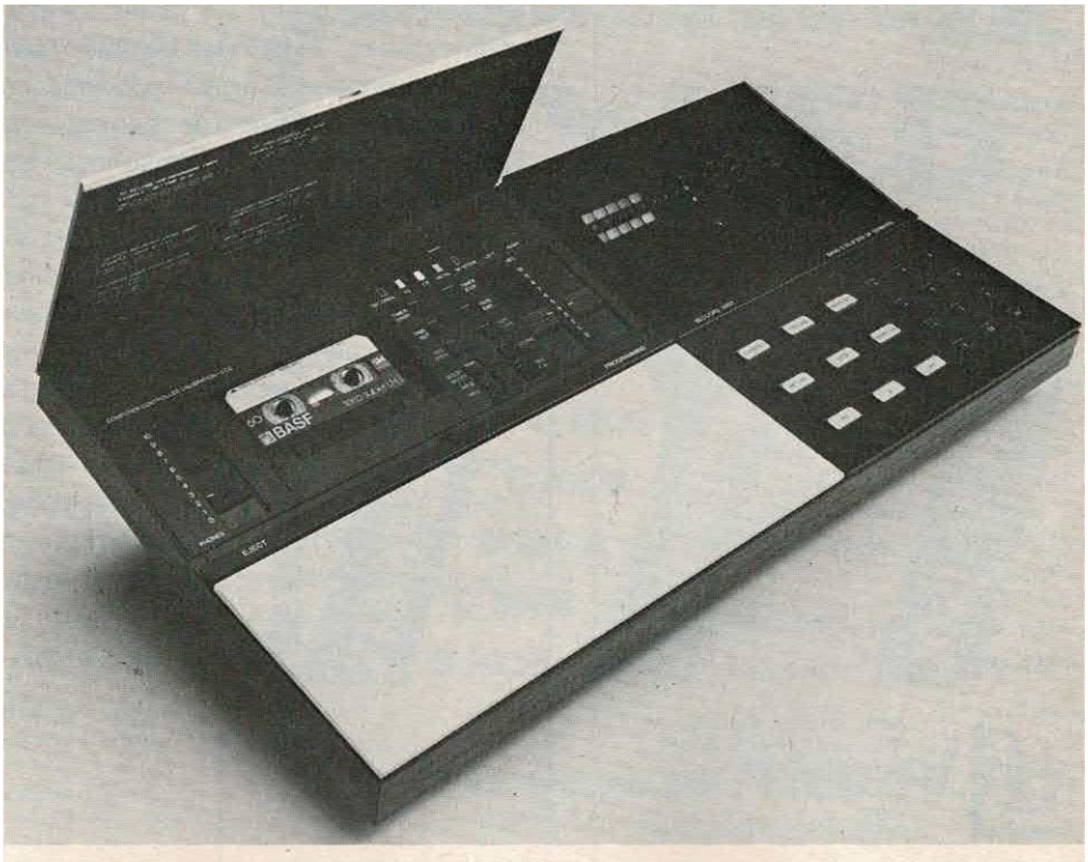

Fig. 1. Beocord 9000.

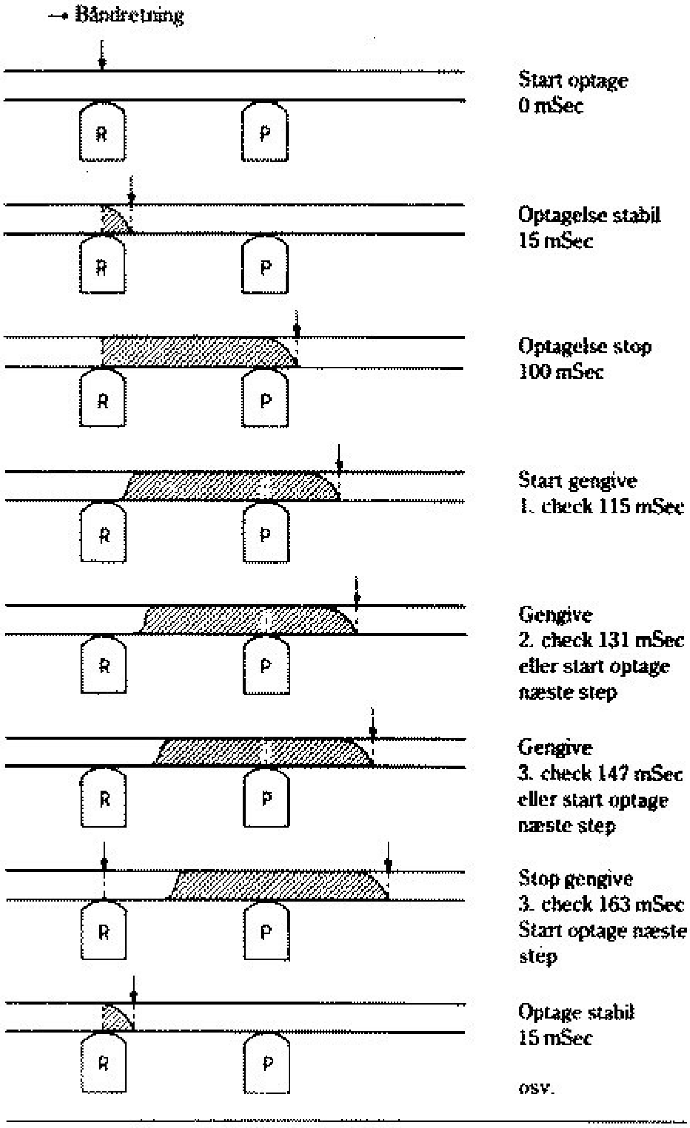

Fig. 2. Båndtypetilpasning.

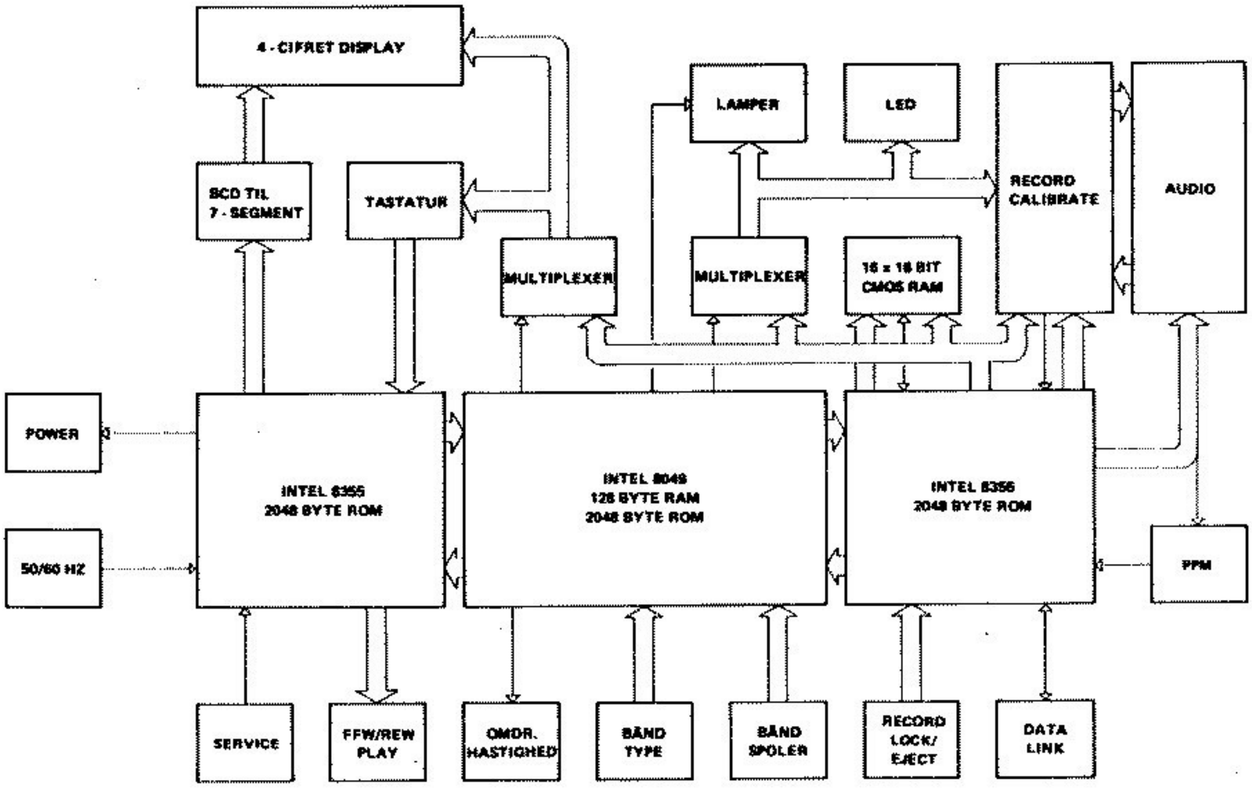

Fig. 3. Beocord 9000. Mikrodatamat med periferikredse.