9.1 GENERAL

The programmed channel is the basic PTS I/O facility and is a standard part of all systems. Its function is to control the flow of data between peripheral channel units and the CPU.

It is also used as the initialization path between CPU and I/O processors, if present in the system. In all uses of the programmed channel, the CPU is the controlling unit.

9.2 I/O INSTRUCTIONS

Data is transferred via the G.P. bus under the control of a program written using the following instructions, which may only be issued in system mode.

* CIO Control Input/Output. Start or Stop an I/O operation.
* INR Input to Register. Transfer one word or character from a Channel Unit buffer to a CPU register.
* OTR Output from Register. Transfer one word or character from a CPU Register to a Channel Unit buffer.
* SST Send Status. A Channel Unit status word is set in a CPU register.
* TST Test Status. Test whether a Channel Unit is busy.

The transfer of each word or character requires a separate instruction and so program loops are used to transfer blocks of data. The programmed channel has two modes of operation, namely Inhibit (or Wait) Mode, and Interrupt Mode.

9.3 CHANNEL UNIT STATES

Once a Channel Unit has been activated by the programmed channel, it is capable of transferring data to or from a peripheral. Processing may continue while I/O is in progress between Channel Unit and device. The following description is applicable to a CHLT or CHRT. Details of the operation of other channel units may be found in the appropriate System Engineering manual.

As shown in figure 9.1, a Channel Unit may be in one of four possible states, as follows:-

* Inactive The unit is idle, and the only instruction it can accept in this state is CIO. Switching to Execute state will occur on reception of CIO Start.

* Execute The unit is operable and transfers data from a peripheral device to a buffer in the Channel Unit, or vice-versa. Unless an error occurs, or a CIO Halt is received, switching will occur to Exchange state on completion of Execute for input. For output, the Channel Unit remains in Execute state.

* Exchange The unit is operable and generates an interrupt to the CPU when this state is entered. The interrupt handler responds with an INR, and data transfer takes place from a Channel Unit buffer to a CPU register. On completion of Exchange a switch will occur to Execute state, unless CIO Halt is received, or an error is detected.

* Wait (For Send Status). An interrupt to the CPU is generated on entry to this state. Wait state is entered from Execute or Exchange states following a CIO Halt, or an error condition. The interrupt handler responds to the interrupt with an SST instruction, and switching then occurs to Inactive state.

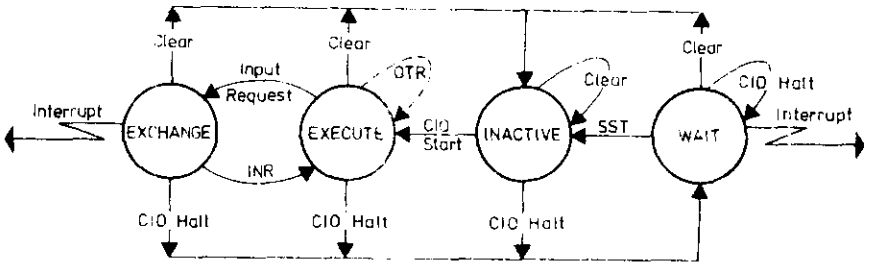

Figure 9.1. Channel Unit States and State-switching.

9.4 PROGRAMMED CHANNEL, INHIBIT (OR WAIT) MODE

In Inhibit Mode the instructions necessary to perform the required I/O are included in the program which needs that I/O. Interrupts are disabled in this mode.

Figure 9.2 depicts a possible I/O sequence in Inhibit mode. Events proceed from left to right in the diagram. Assume that the Channel Unit is in Inactive state at the start of the sequence, e.g. after power-on.

1. The program issues a CIO Start instruction and then waits. This causes the Channel Unit to switch into Execute state and transfer a word from the relevant Selector Unit to the Channel Unit buffer. Exchange state is then entered so that the transfer can be completed. An interrupt is sent to the CPU but is not acted upon because interrupts are inhibited.

2. The program now issues an INR instruction which, when accepted by the Channel Unit in Exchange state, causes a word or character to be transferred from the Channel Unit buffer to a Register in the CPU. After execution of the INR, the Channel Unit switches to Execute state and transfers another word or character, if available, from device to Channel Unit buffer. A switch is made back to Exchange state on completion of Execute, and another interrupt is generated but not acted upon.

3. Repeat (2).

4. The program has now performed all the input transfers it requires and issues a CIO Halt instruction. This switches the state of the Channel Unit from Exchange to Wait, and again a non-actioned interrupt is generated. An SST instruction in the program now switches the Channel Unit back to Inactive state ready for the next CIO Start instruction.

5. The program has now obtained the data it requires and may continue processing.

An important point to notice about Inhibit mode is that the CPU is forced to wait while I/O operations between Channel Unit and device are in progress, and so it is idle for that time. This can be largely overcome by exploiting the interrupt system and using the other mode of the programmed channel, namely Interrupt Mode.

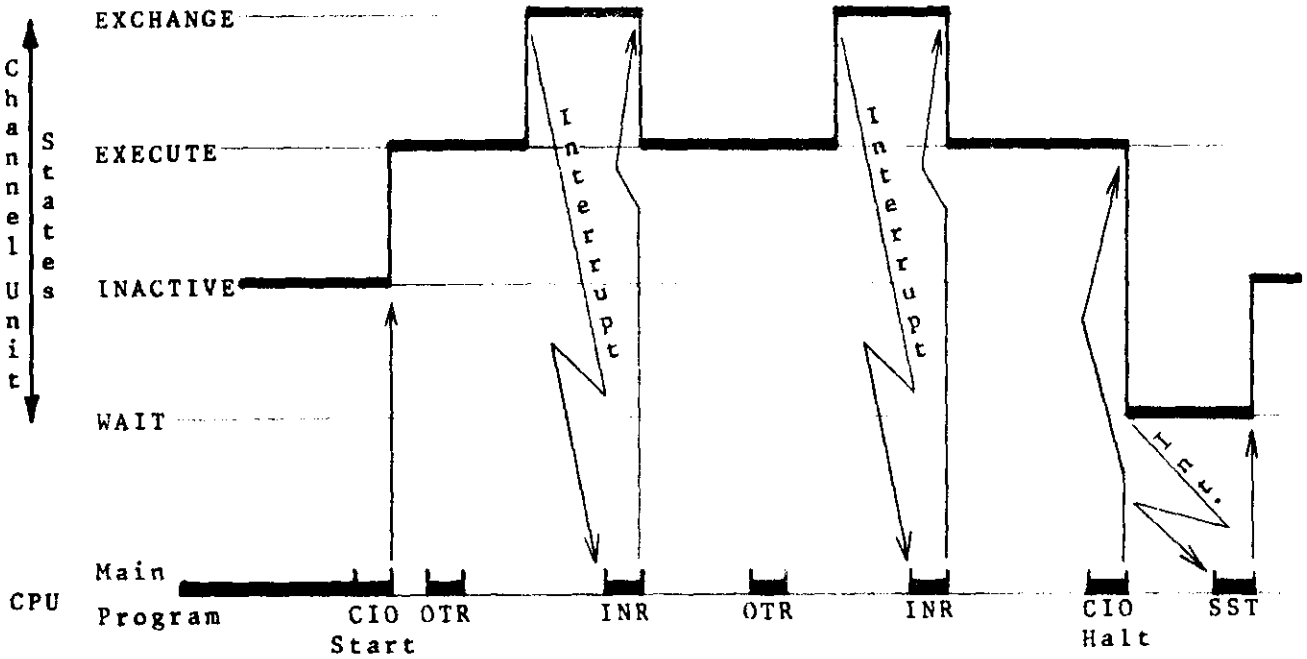

Figure 9.2. Programmed Channel, Inhibit Mode Sequence.

9.5 PROGRAMMED CHANNEL, INTERRUPT MODE

Unlike Inhibit Mode, the instructions required for I/O in this mode are not included in the Main program, but are part of a special Interrupt Routine, invoked by interrupt signals raised by the Channel Unit hardware. Interrupts must, of course, be enabled.

1. The program issues a CIO Start instruction and continues with processing. The Channel Unit switches into Execute state and a word can be transported from a peripheral device to the Channel Unit buffer. After receipt of the word a switch is made to Exchange state, and this generates an interrupt to the CPU.

2. Control passes from the main program to the interrupt routine, which issues an INR instruction. The Channel Unit, in Exchange state, accepts this and transfers the word or character from the Channel Unit buffer to a CPU register. The main program resumes processing after the INR is completed. The Channel Unit switches to Execute state and transfers another word from device to CU buffer, whereupon a switch is made back to Exchange state, and another interrupt is generated.

3. Repeat 2.

4. All required input has now been performed, and the interrupt routine issues a CIO Halt instruction. The Channel Unit switches from Exchange to Wait state and generates another interrupt to the CPU. The main program is suspended and the interrupt routine gains control and issues an SST instruction which switches the Channel Unit back to Inactive state.

5. The main program resumes processing until it requires the data, which is ready to be used in memory at this point.

This example assumes that the program is able to continue processing while the data is being retrieved. If the I/O had not been completed before it was required by the program, it would have been necessary to include coding so that the program would wait for completion.

It is important to note that, unlike Inhibit Mode, the main program is not held up waiting for the Channel Unit. As long as there is processing to be done, it can carry on simultaneously with Channel Unit/device action, except for short interruptions by the interrupt routine which contains the I/O instructions.

Note that upon CIO Start for output, the Channel Unit switches from Inactive to Exchange, not Execute. This is because, for output, the first part of the transfer path is from CPU to Channel Unit buffer, whereas for input, it is from device to Channel Unit buffer.