Chapter 4

The role and the mechanism of Programmable Pulse Generator for NMR Imaging

In MMR imaging system, each method of image reconstruction requires the different excitation sequences (see chapter3), involving various pulse shapes with different timing. In our case, the shape of each pulse can be first drawn by the computer and then send to be stored in the memory of the pulse shaping unit. The relative delay times between these shaped pulses, depending on imaging methods are controlled by the pulse programmer, which in turn can be programmed by the computer. So the pulse programmer and the pulse shaping unit are the instruments that originate the rf and field excitation sequences to the sample and these excitation sequences can be programmed by the computer. The diagram of NMR imaging system using a pulse programmer and the four pulse shaping units is illustrated in Fig.4-1.

NMR Imaging system

Refering to Fig.4-1, the computer sends the excitation sequence, by passing the GPIB, to be stored in the pulse programmer and the pulse shaping units. When the system is ready to be executed, the computer will send the starting signal, by passing the GPIB, to the pulse programmer. Then the pulse programmer will send sequential trains of pulses to the four pulse shaping units according to programmed times. The x-, y- and the z-pulses will be further amplified and sent to the x-, y- and z-gradient coils, respectively. The other of the shaped rf-pulse will be used to modulate the rf signal, by the rf modulator; then further amplified by the rf power amplifier to be delivered to the rf coil around the sample. At this time, the sample received the encoding energy from rf coil, magnet, x-, y-, and z-gradient coils; consequently, it will release the nmr signal which can be analysed to relate the position and the density of sample to its frequencies and intensities in rf range. The amplitude of the signal is minute, so it will need to be amplified by rf preamplifier and rf amplifier before the be comparison with the reference signal of rf oscillator by the phase sensitive detector. The output signal of a phase sensitive detector, in af range, will be stored in the signal averager before be transfered to the computer. Finally, the computer will being transfered do the compution and reconstruct the sample image.

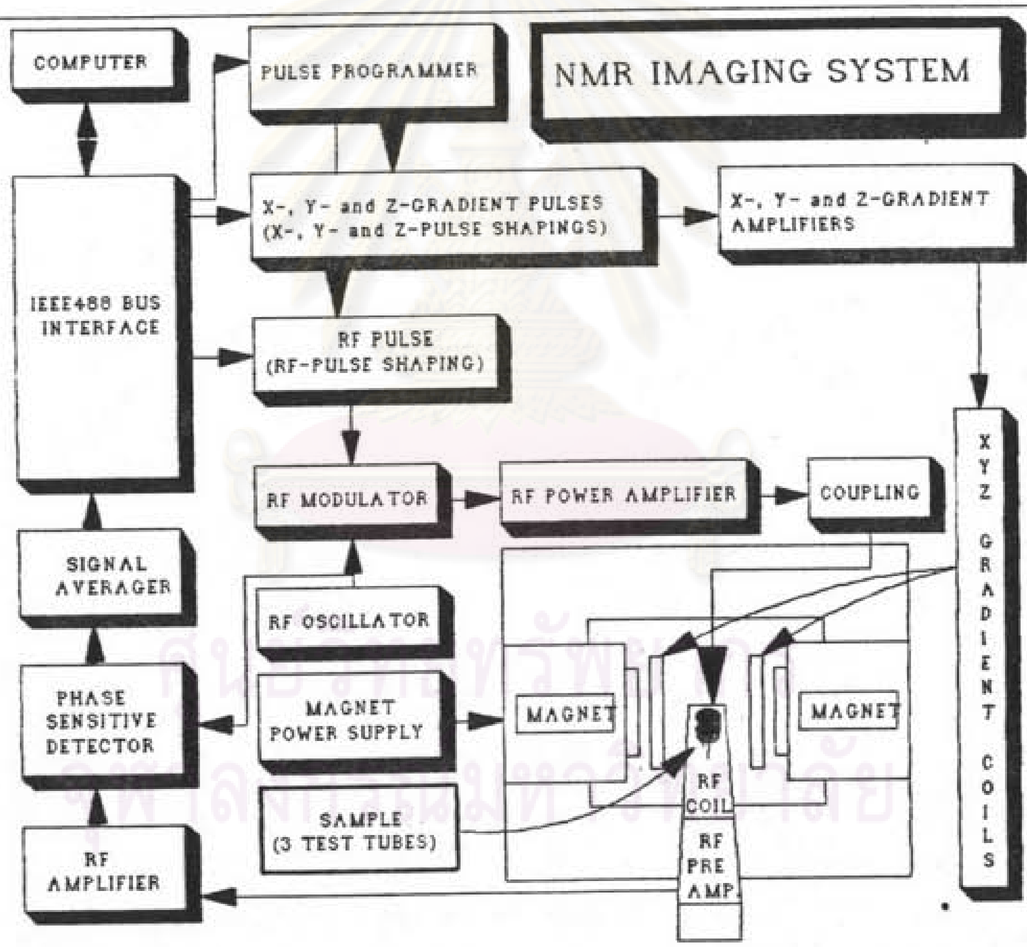


Fig.4-1 The diagram of NMR imaging system containing a pulse programmer and pulse shaping units.

The function of Programmable Pulse Generator

This thesis has built a programmable pulse generator that consist of the pulse programmer and the pulse shaping units. To control the excitation sequence, the aim of these instruments, we provide the pulse programmer to control the time between the adjacent pulses in the same sequential trains, and the pulse shaping to control the shapes of those pulses. This is the main idea of programmable pulse generator. For example, if we want the excitation sequence that shown in Fig.4-2(a), we will program the trigger pulse train on the pulse programmer and the sequential pulse shapes on the pulse shaping units as shown in Fig.4-2(b) and Fig.4-2(c), respectively.

The required excitation sequence for a set of valid data starts from the maximum positive amplitude of the y-pulse shape (for Gy) then decrease in steps by picking the next stored pulse shape in the following cycle until the maximum negative amplitude of y-pulse shape. In practice we will repeat these pulse trains as many times as need for the matrix, for example 32 times for 32×32 dot image or 64 times for 64×64 dot image, etc. The data matrix from these excitation sequences will be used to reconstruct the image by two-dimensional Fourier transformation (see detail in chapter 3). The pulse programmer is programmed to provide the trigger pulses to various channels; it is capable of having 8 channels but at present need only 4 channels, they are for x-, y-, z-, and rf-pulse trains. The trigger pulses are made to sent to x-, y-, z-, and the rf-pulse shaping units, respectively, with the programmed timing. Then, in turn, the respective pulse shaping units will deliver the stored waveforms out to the other devices, such as x-, y-, z-amplifiers and rf modulator. Each time after a trigger pulse the pulse shaping unit continute to give output of the stored shape until meeting with the stop code. In case that it is stopped by the end of sequential pulse shapes, the pulse shaping of this sequential pulse shapes will be recycle to wait the next trigger pulse from the pulse

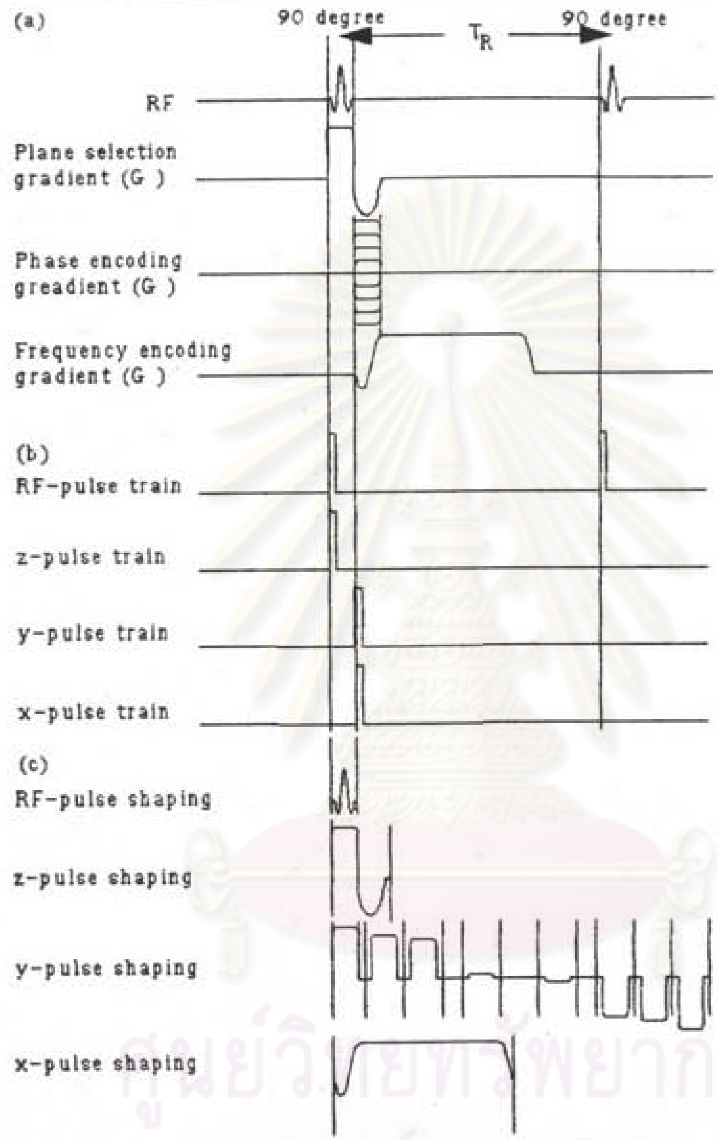


Fig.4-2 The concept of programmable pulse generator (a) The excitation sequence that we want. From y-sequential train illustrated that this sequence will be executed repeatedly until end of y-sequiential train. (b) The x-, y-, z- and RF-trigger pulse trains that were programmed on pulse programmer and sent to x-, y-, z- and RF-pulse shapings, respectively. (c) The pulse shapes that were programmed on pulse shaping units. The vertical dashed lines represented the stop code to wait the trigger pulse from pulse programmer.

programmer again, so rf-pulse shape in Fig.4-2(c) is programmed with only one pulse shape, but the rf-pulse shaping received the two trigger pulses in the one pulse train. The other case that it is stopped by the stopping code, the pulse shaping of this sequential pulse shapes will wait the next trigger pulse from the pulse programmer and if the next trigger pulse arrived, the next pulse shape will be sent out until reaching the stopping code again or the end of the sequential pulse shapes. So y-pulse shaping in Fig.4-2(c) is programmed with many pulse shapes having the stop codes, inserted between the adjacent pulse shapes (in the figure represented by the vertical dashed line). The recycle of pulse trains are acheived by the trigger signal from the computer through the GPIB to the pulse programmer until reaching 32 times for 32×32 dot image or 64 times for 64×64 dot image.

The mechanism of the pulse programmer (PPM)

Since the pulse programmer and the pulse shaping unit concern with the timing which is of importance, so that the adjustable frequency is needed to be used as the time-base. The mechanism for the pulse programmer to retain accurate time in each interval of times between the adjacent trigger pulses is to count the square pulse train, that has constant frequency from a time-base coincide with the number stored in the pulse programmer's memory. Then it will send a trigger pulse to the next selective channel. The block diagram of the pulse programmer is shown in Fig.4-3.

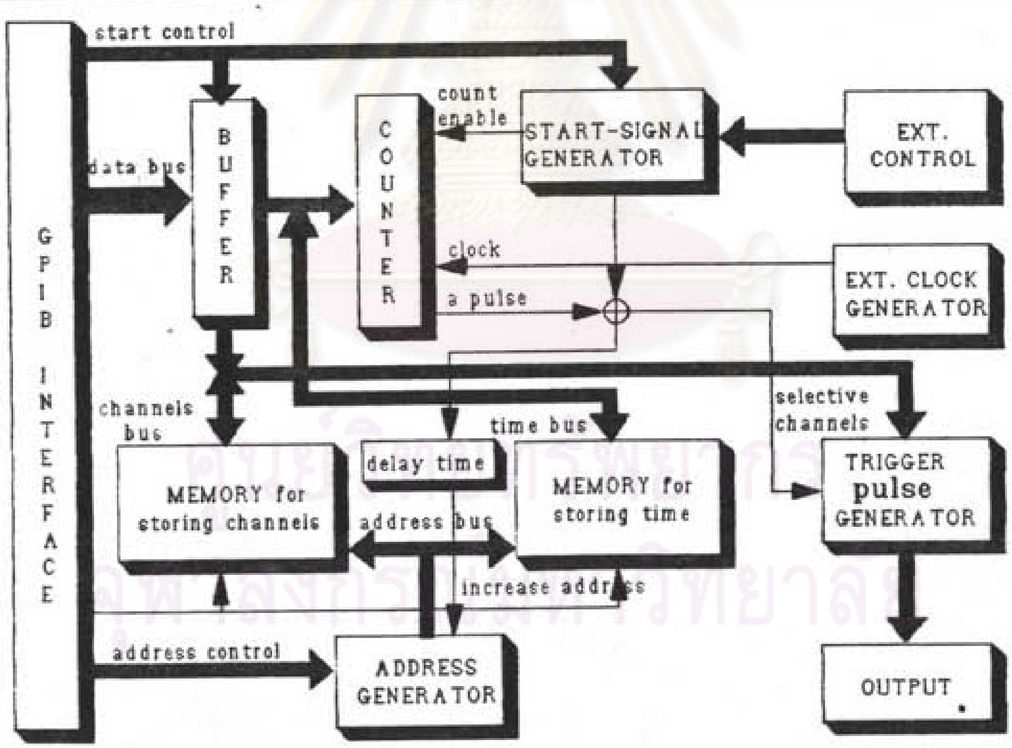
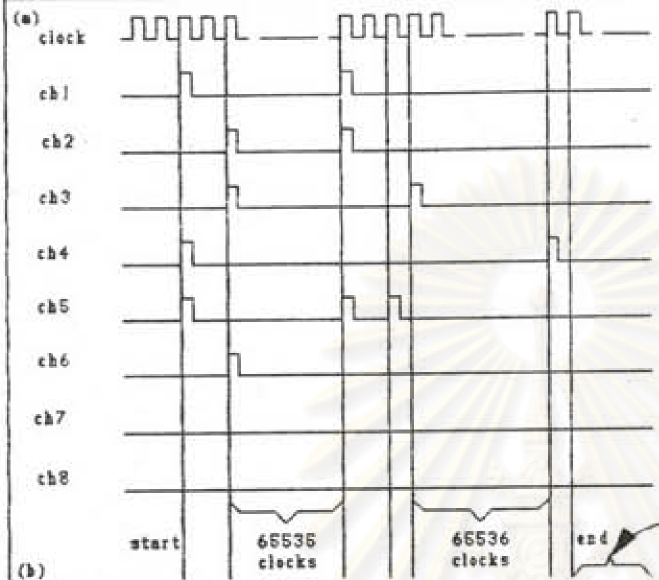


Fig.4-3 The block diagram of working process of the pulse programmer.

The information storing of the pulse programmer

From Fig.4-3, the computer can send the informations of pulse trains to be stored in the memory of the pulse programmer by passing the GPIB interface which is controlled by 8035 microprocessor. This microprocessor is used to emulate to the GPIB listener and to control the information storing in the pulse programmer. When informations from the computer arrived, the GPIB interface will control the buffer, address generator and both memories - memory for storing channels and memory for storing time - to keep those informations in both memories. The other parts is disabled at this time. At the end of this process the computer send the ending codes which are the three-byte zero to the pulse programmer. Then the GPIB interface will set all parts of the pulse programmer to enable to be ready to start the executing process. In the information storing, the memory for storing channels, a 6116 RAM is used to keep the state of 8 channels and for the memory for storing time, two 6116 RAMs are used to keep the time interval between adjacent pulses - one is used to keep the low order byte of time interval and the other is used to keep the high order byte of time interval. The timing informations stored in the pulse programmer are the time interval between two different-channel pulses or between two pulses of the same channel whatever is the shorter time (see Fig.4-4).

If this shorter time interval is more than 65535, that is the maximum value of memory in one address, it will split into two or more addresses as shown in Fig.4-4 (address 4 and address 5).



(b)

channels									
channels	hex	19	26	13	1	4	0	8	0
	dec	25	38	19	16	4	0	8	0
low order time		2	255	2	1	255	1	1	0
high order time		0	255	0	0	255	0	0	0
address		0	1	2	3	4	5	6	7

Fig.4-4 The informations that stored on pulse programmer by the low and high order time keep number of clocks from time-base, the maximum value is 65536 clocks. If it is more than this value, it is used to spliting into two or more than one address. (a) The sequential pulse trains on the outputs of each channel of pulse programmer that we want. (b) The informations that store on pulse programmer give the sequential pulse trains as we wish.

The executing process of the pulse programmer

After the information storing in the pulse programmer, both memories are in the read state and the other parts of the pulse programmer are enabled. The system is ready to be executed; it is waiting to receive the trigger signal from the external control or the trigger command of the GPIB from the computer. At this time, the channel bus and the time bus have the values of both memories at address 0. When the trigger signal or the trigger command arrived, the starting-signal generator will control the counter to load the time interval from the time bus and give a pulse to the trigger pulse generator . The output of the trigger pulse generator will trigger channels upto 8 channels, depending on the values on the channel bus at that time. At the same time, the ripple clock is provided to increase address on the address generator by delaying it a little time in order to give enough time for the counter to load the old time interval from the time bus. So the new channels and the new time interval from both memories at the next address are released on the channel bus and the time bus, respectively.

When the counter has already loaded the time interval from the time bus, it will continue to count this value down by using the clock from external clock generator until it is zero. Then the counter will also generate the pulse to the trigger pulse generator and to increase address on the address generator by delaying it a little time. It will repeat as the above description until end of the stored information. After then address generator will be reset automatically to zero and the pulse programmer ready to be executed again by waiting to receive the trigger signal from the external control of the trigger command from the computer.

If the trigger signal or the trigger command arrive during the executing time, it will be ignored. It has the GPIB commands that can control the pulse programmer to ignored the trigger signal from the external control (see the GPIB commands in appendix A). On the front panel key operation of the pulse programmer there is a button that can control the start-signal generator. When this button is pressed, the pulse programmer will be executed in a 'rerun' mode - executing in repeatition until this button is pressed again. It is very useful for checking the sequential pulses that were programmed. It can be controlled to not work by the GPIB commands also. The details of the pulse programmer will be considered in chapter 5 again.

The mechanism of the pulse shaping unit (PSP)

The PSP is the instrument that require the adjustable constant frequency, it should be the same source that contribute to the pulse programmer. This frequency is a factor that change the stored shapes along the time axis. These shapes are stored in the pulse shaping's memory that size is 8K RAM for IC 6264 or 32K RAM for IC 62256. Between the adjacent shapes there should be a stop code that is the ASCII (255). When the pulse shaping unit read this code from its memory it gives zero voltage output. It is the same as the pulse shaping unit read the ASCII (127) from its memory and the pulse shaping will stop the executing process. If the pulse shaping unit reads the ASCII (0) from its memory, its gives the negative maximum voltage output. And if it read the ASCII (254) from its memory, its gives the positive maximum voltage output. For the other ASCII that the pulse shaping unit reads from its memory, its output is linear proportion to the different value between the positive maximum value and the negative maximum value. The block diagram of the pulse shaping unit is shown in Fig.4-5.

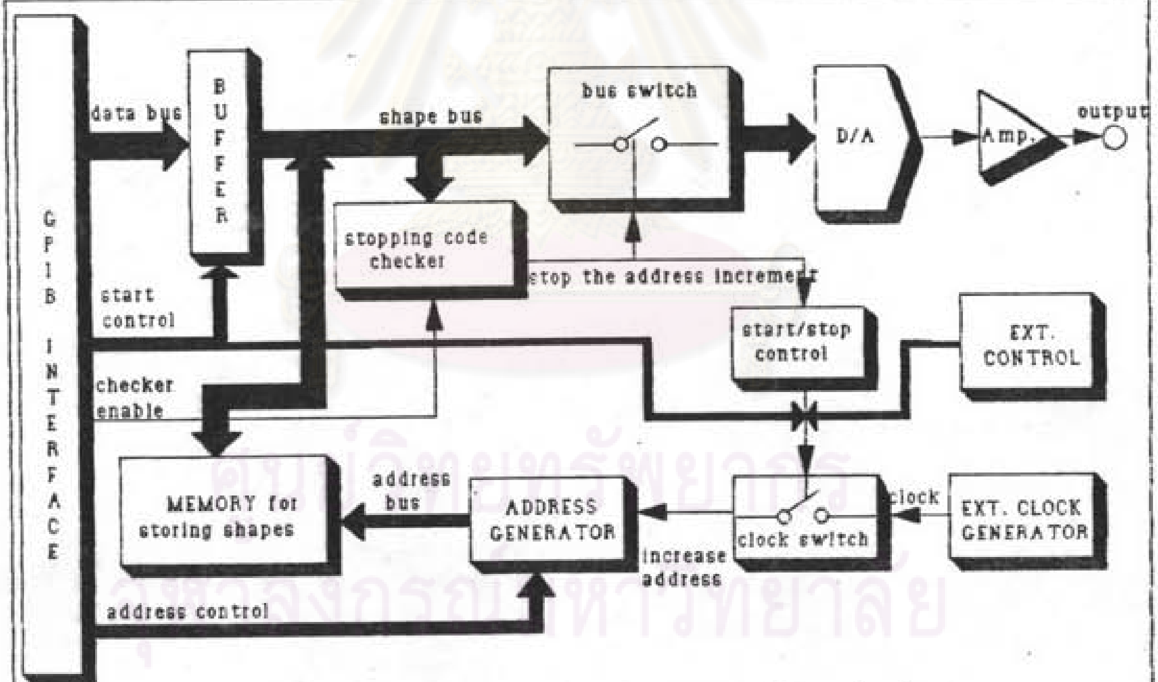
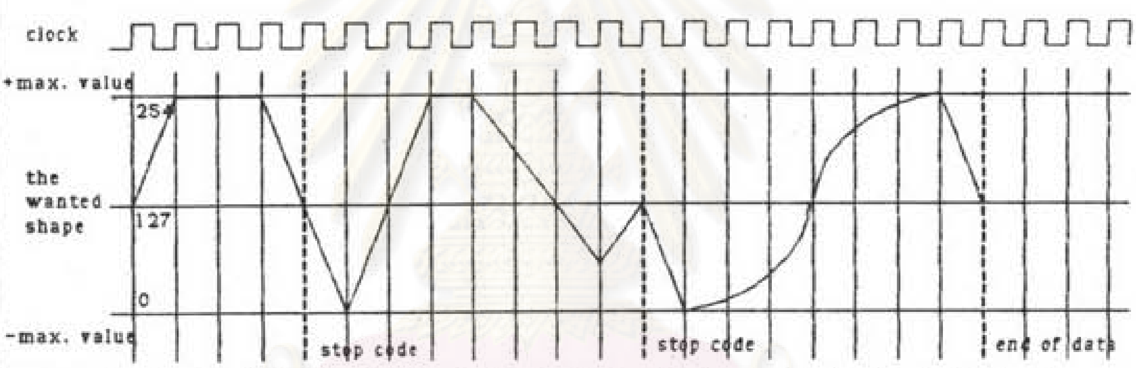


Fig.4-5 The block diagram of working process of the pulse shaping.

The Information storing of the pulse shaping unit

From Fig.4-4, the computer can send the informations of the shape to be stored in the pulse shaping unit by passing the GPIB interface, in the same manner as one for the pulse programmer. When the information from computer arrive, the GPIB interface will control the buffer, address generator and memory for storing shapes to keep those informations in the memory. The other parts is disabled during this time. At the end of this process, the computer send the ending code, which is the two-byte ASCII (255), to the pulse programmer; then the GPIB interface will set all parts of the pulse shaping to enable to prepare to start the executing process. The informations that are stored in the pulse shaping unit are considered from the sampling points on the required shapes as shown in Fig.4-6.



shape	127	254	254	254	255	0	127	254	254	190	127	63	255	0	17	63	127	190	237	254	255	255
address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

the ending code

Fig.4-6 The information that stored on pulse shaping. (a) The shapes on the output of pulse shaping that we want. (b) The informations that stored on pulse shaping give the shape as we want.

In practice, there cannot be the two sampling points at the same time, so the required shapes in Fig.4-6 cannot be the vertical line.

The executing process of the pulse shaping unit

After the information have been stored in the pulse shaping unit. Its memory is in the read state and the other parts of the pulse shaping are enabled. The system ready to be executed; it is waiting to receive the trigger signal from the external control or the trigger command of the GPIB from the computer. At this time, the shape bus has the value of memory at address 0; this value will be converted to the analog signal by digital to analog converter (D/A) and amplified to the output. In order to eliminate the output signal to zero voltage, the first address of the first shape (address 0) should be the ASCII (127). When the trigger signal or the trigger command arrived, the start/stop control will turn the clock switch and the bus switch on. Then the external clock can pass through the address generator that consists of the up counter and the down counter. When the clock passes one period, the address bus is increased with one address and the new value of the new address is converted and amplified to the output too. The shape that has been stored in the pulse shaping's memory will remain to appear on the output until the value on the shape is the ASCII (255), that is the stop code. The stop code checker can detect this code; it will send the signal to turn the bus switch off and to turn the clock switch off by passing the start/stop control. Then it is no clock to the address generator, so it will remain the old address of the ASCII (255) until the trigger signal from the external control or the trigger command of the GPIB arrived again.

The trigger signal or the trigger command cannot work, when they arrive on the executing time. It has the GPIB commands that can control the pulse programmer to ignored the trigger signal from the external control (see the GPIB commands in appendix A). On the front panel key operation of the pulse shaping unit, it has a button that can control the start/stop control. When it is pressed, the pulse programmer will be executed in 'single run' mode to release one shape on the output. It can be controlled to not work by the GPIB commands also. The details on building of the pulse programmer will be considered in chapter 6 again.