TONE GENERATOR

BY

SUBATHRA S

TONE GENERATOR

AIM

To wite an assembly language program to generate tones using software.

APPARATUS REQUIRED

- 8085 Microprocessor Trainer kit
- Flat Ribbon Cable
- Tone generator kit
- Power Supply

DESCRIPITION

Tone generator is a circuit which generator different tones depending on the input signal and its frequency. This circuit contains AND gates (IC is 74HC102).The two inputs for AND gates are one from 8255 (26 core FRC cable connected to P3) and other input is from 8253 IC on the kit. With the help of FRC, we can connect this signal from p2 on the kit to JP2 On the interface. These input signal are modulated and outputted to the transistor Q1 whose collector current varies depending on the base current fed by the O/P of the AND gate, this variation in collector current causes the speaker diaphragm to vibrate at different frequencies and hence to output various tones.

The user can write various program to generate different tones to the output of speaker.

On the interface one beard speaker is mounted and also to work under silent areas headphone socket is provided in addition to this, a volume control pit R6(500 ohm) provided to vary the volume of the output.

Steps to be followed to generate tone using software

Calculate the Hexa count to be loaded to 8253 16 bit counter for the particular frequency of the required time. Using the formula,

Count (dec) =1.5 MHz / (frequency required)

After this convert the count in decimal value to Hex value & load 16 bit data to 8253 counts. In hardware of the interface when you connect FRC clock1 of 8253 is shorted to clock2 of the same on the kit, at clock2 you have 1.5MHz of the frequency.

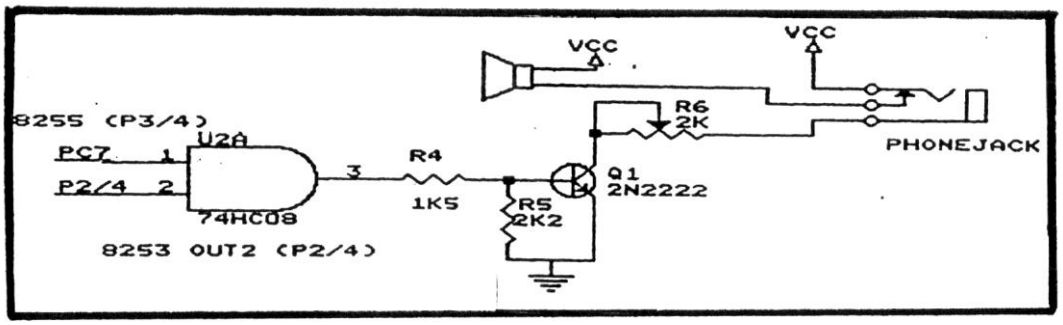
a) Disable the output of AND gate by sending ‘0’ to PC7
b) Enable the same by sending1 to PC7
c) Call different delays to generate tones for that particular duration
d) Disable AND gate output and call delay in order to give pause between tones
e) Load next 16 bit hex data to 8253 counter and repeat the above steps

Some of the fundamental frequencies are given below

TONE NAME	OCTAVE 0	OCTAVE 1	OCTAVE 2	OCTAVE 3
Sa	130.810	**261.630**	523.250	1046.500
Re	146.830	**293.660**	587.330	1174.700
Ga	164.810	**329.630**	659.260	1318.500
Ma	174.610	**349.230**	698.460	1396.900

Pa	196.000	**392.000**	784.000	1568.000
Dha	220.000	**440.000**	880.000	1760.000
Ni	246.940	**493.883**	987.770	1975.540

CIRCUIT DIAGRAM



TO WORK WITH TONE GENERATOR

a) In addition to the FRC connected from P3 of kit to JP1 on interface(26 pin). Connect one more FRC connect P2 (20 pin) on the kit to JP2(20 pin) on the interface.

b) Connect +5 & GND to generate low, execute the program as GO <starting address> <EXEC> . Now you can hear music (or tone) at the output of the speaker depending on your program. The tone volume can be controlled by varying R6 pot mounted on the interface.

If you connect head phone externally to jack provided the on board speaker is disabled and you will get the output on head phone.

FUNDAMENTAL FREQUENCY

OCTAVE — 1

SA = 1.5 Mhz / 261.630 = 5740.5 = 5741_d = $166D_H$
RE = 1.5 Mhz / 293.660 = 5107.9 = 5108_d = $13F4_H$
GA = 1.5 Mhz / 329.630 = 4550.5 = 4551_d = $11C7_H$
MA = 1.5 Mhz / 349.230 = 4295.1 = 4295_d = $10C7_H$
PA = 1.5 Mhz / 392.000 = 3826.5 = 3827_d = $0EF3_H$
DHA = 1.5 Mhz / 440.000 = 3409.09 = 3409_d = $0D51_H$
NI = 1.5 Mhz / 493.883 = 3037.15 = 3037_d = $0BDD_H$

ASSEMBLY LANGUAGE PROGRAM

ADDRESS	LABEL	MNEMONICS	OPCODE/OPERAND
C100		**MVI A,80_H**	3E 80
C102		**OUT CWR**	D3 DB
C104		**MVI A,$B6_H$**	3E B6
C106		**OUT TIMERCTRL**	D3 CB
C108	**REPEAT**	**MVI B,07_H**	06 07
C10A		**LXI H,$C200_H$**	21 00 C2

C10D	**NEXT**	**MVI A,M**	7E
C10E		**OUT TIMER2**	D3 CA
C110		**INX H**	23
C111		**MOV A,M**	7E
C112		**OUT TIMER2**	D3 CA
C114		**MVI A,80_H**	3E 80
C116		**OUT PORTC**	D3 DA
C118		**CALL DELAY**	CD 27 C1
C11B		**MVI A,00_H**	3E 00
C11D		**OUT PORTC**	D3 DA
C11F		**INX H**	23
C120		**DCR B**	05
C121		**JNZ NEXT**	C2 OD C1
C124		**JMP REPEAT**	C3 08 C1
C127	**DELAY**	**MVI C,04_H**	0E 04
C129	**START**	**LXI D,$FFFF_H$**	11 FF FF
C12B	**LOOP1**	**DCX D**	1B
C12C		**MOV A,E**	7B
C12D		**ORA D**	B2
C12E		**JNZ LOOP1**	C2 2B C1
C131		**DCR C**	0D
C132		**JNZ START**	C2 29 C1
C135		**RET**	C9

PROGRAM TRACE

<table>
<tr><th>LABEL</th><th>MNEMONICS</th><th>DESCRIPTION</th></tr>
<tr><td></td><td>MVI A,80_H</td><td>Initializing the ports of the PPI 8255 as O/P ports by writing the control word as 80_H.

<table>
<tr><td>DATA BITS</td><td>D_7
1</td><td>D_6 D_5
0 0</td><td>D_4
0</td><td>D_3
0</td><td>D_2
0</td><td>D_1
0</td><td>D_0
0</td></tr>
<tr><td>COMMENT</td><td>I/O mode</td><td>Mode0</td><td>PortA O/P</td><td>PortC Upper O/P</td><td>Mode0</td><td>PortB O/P</td><td>PortC Lower O/P</td></tr>
</table>
80_H is moved to accumulator.
REGISTERS
<table>
<tr><td>A</td><td>80</td><td>XX</td><td>F</td></tr>
<tr><td>B</td><td>XX</td><td>XX</td><td>C</td></tr>
<tr><td>D</td><td>XX</td><td>XX</td><td>E</td></tr>
<tr><td>H</td><td>XX</td><td>XX</td><td>L</td></tr>
</table>
</td></tr>
<tr><td></td><td>OUT CWR</td><td>Control word specifies the I/O function for each port of 8255.</td></tr>
<tr><td></td><td>MVI A,$B6_H$</td><td>Initializing the COUNTER 2 of the PIT 8253 in MODE 3 by writing the control word as $B6_H$.

<table>
<tr><td>COMMENT</td><td colspan="2">SELECT COUNTER</td><td colspan="2">READ/ LOAD</td><td colspan="3">MODE</td><td>BCD/ BINARY COUNT</td></tr>
<tr><td></td><td>SC1</td><td>SC0</td><td>RL1</td><td>RL0</td><td>M2</td><td>M1</td><td>M0</td><td></td></tr>
<tr><td>DATA BITS</td><td>D_7
1</td><td>D_6
0</td><td>D_5
1</td><td>D_4
1</td><td>D_3
0</td><td>D_2
1</td><td>D_1
1</td><td>D_0
0</td></tr>
<tr><td>OBSERVE</td><td colspan="2">C0UNTER 2</td><td colspan="2">LSB/MSB</td><td colspan="3">MODE 3</td><td>BINARY COUNT</td></tr>
</table>
</td></tr>
</table>

		$B6_H$ is moved to accumulator. **REGISTERS** **A** **B6** XX F B XX XX C D XX XX E H XX XX L NOTE: ❖ 8253 IS USED TO GENERATE ACCURATE TIME DELAY FOR THE SQUARE WAVE IN MODE3. ❖ AT THE END OF COUNT, A PULSE IS GENERATED WHICH INTERRUPT THE MICROPROCESSOR.
	OUT TIMERCTRL	Output it through TIMERCTRL. NOTE: ❖ Opcode of TIMERCTRL is CB.
REPEAT	**MVI B,07_H**	Initialize B register with number of count. REGISTERS A XX XX F **B** **07** XX C D XX XX E H XX XX L
	LXI H,$C200_H$	Initialize the memory pointer at $C200_H$ .i.e. loads the 16-bit data in the register pair designated. **REGISTERS** A XX XX F B 07 XX C D XX XX E **H** **C2** **00** **L** $C200_H$ is the memory pointer. **MEMORY** **C200** **CB** C201 2C C202 E7 C203 27 C204 8D
NEXT	**MVI A,M**	LSB of 'Sa' is loaded in to accumulator. **REGISTERS** **A** **CB** XX F B XX XX C D XX XX E H XX XX L
	OUT TIMER2	CB_H is outputted thro TIMER2. NOTE: ❖ Opcode of TIMER2 is CA.
	INX H	Increment the HL register pair by 1.The instruction views the contents of the HL registers as a 16-bit number. No flags are affected.

		REGISTERS A CB XX F B 07 XX C D XX XX E **H C2 01 L** $C201_H$ is the memory pointer. **MEMORY** C200 CB **C201 2C** ← HL memory pointer C202 E7 C203 27 C204 8D
	MOV A,M	MSB of 'Sa' is loaded in to accumulator. **REGISTERS** **A 2C** XX F B 07 XX C D XX XX E H C2 01 L
	OUT TIMER2	$2C_H$ is outputted thro TIMER2.
	MVI A,80_H	Move 80_H immediately in to accumulator. **REGISTERS** **A 80** XX F B 07 XX C D XX XX E H C2 01 L
	OUT PORTC	Initialize the ports. DATA | **PC_7** | PC_6 | PC_5 | PC_4 | PC_3 | PC_2 | PC_1 | PC_0 BITS | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 **PC_7 =1** ,Enable AND gate output by sending 1 bit (high signal) to **PC_7.**
	CALL DELAY	Call delay sub program.
	MVI A,00_H	Clear the accumulator. **REGISTERS** **A 00** XX F B 07 XX C D XX XX E H C2 01 L
	OUT PORTC	Initialize the ports. DATA | **PC_7** | PC_6 | PC_5 | PC_4 | PC_3 | PC_2 | PC_1 | PC_0 BITS | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 **PC_7 =0** ,Disable AND gate by sending 0 bit (low signal) to **PC_7**

	INX H	Increment the HL register pair by 1. **REGISTERS** **A** **E7** XX F B 07 XX C D XX XX E H C2 02 L
	DCR B	Decrease the count of B register since 'Sa' is obtained. **REGISTERS** **A** **XX** XX F B 06 XX C D XX XX E H C2 02 L
	JNZ NEXT	For single loop, to play sa, re, ga, ma, pa, dha, ni Jump to next.
	JMP REPEAT	In order to play continuously the tune, repeat the process.

DELAY	**MVI C,04_H**	Move 04_H immediately to C register. **REGISTERS** A XX XX F B XX **04** **C** D XX XX E H XX XX L
START	**LXI D,$FFFF_H$**	Initialize the memory pointer at $FFFF_H$ .i.e. loads the 16-bit data in the register pair designated. **REGISTERS** A XX XX F B XX 04 C **D** **FF** **FF** **E** H XX XX L $FFFF_H$ is the memory pointer. **MEMORY** **FFFF** **XX** FFFE XX FFFD XX FFFC XX FFFB XX
LOOP1	**DCX D**	Decrement the HL register pair by 1.The instruction views the contents of the HL registers as a 16-bit number. No flags are affected. **REGISTERS** A XX XX F B XX XX C **D** **FF** **FE** **E** H XX XX L $FFFE_H$ is the memory pointer.

		MEMORY FFFF XX **FFFE XX** ← DE memory pointer FFFD XX FFFC XX FFFB XX
	MOV A,E	Move E register to accumulator. **REGISTERS** **A FE** XX F B XX XX C D XX XX E H XX XX L
	ORA D	OR the accumulator content with D register content.
	JNZ LOOP1	Jump if no zero to Labled LOOP1.
	DCR C	Decrement the C register content. **REGISTERS** A XX XX F B XX **03** **C** D XX XX E H XX XX L
	JNZ START	Jump if no zero to Labeled START.
	RET	Return to main program.

EXECUTION

ADDRESS	TONE NAME	FREQUENCY (Hz)	HEXADECIMAL (hexa)
C200	Sa	**261.630**	1665_H
C202	Re	**293.660**	$13F4_H$
C204	Ga	**329.630**	$11C7_H$
C206	Ma	**349.230**	$10C7_H$
C208	Pa	**392.000**	$0EF3_H$
C20A	Dha	**440.000**	$0D57_H$
C20C	Ni	**493.883**	$0BDD_H$

REFERENCE

1. Ramesh S.Gaonkar, Microprocessor Architecture, Programming, and Applications, Fourth Edition, Penram International Publishing (India), 2000.

2. S.Subathra, “Advanced Microprocessor Laboratory”, Record work, Adhiparashakthi Engineering College, Melmaruvathur, October 2002

3. S.Subathra, “Programming in 8085 Microprocessor and its applications – An Innovative Analysis”, Technical Report, Adhiparashakthi Engineering College, Melmaruvathur, March 2003

4. Micro-85 EB, User Manual, Version – 3.0, CAT #M85 EB-002, VI Microsystems Pvt. Ltd., Chennai.