APRIL 18, 2021

FREQUENCY GENERATOR

GROUP 09

MRUDUL M NAIR
SHIVASHANKAR S MENON
ANAND S
HRITHIK NAMBIAR
JISHNU R WARRIER
ATHUL SHIBU KURIEN

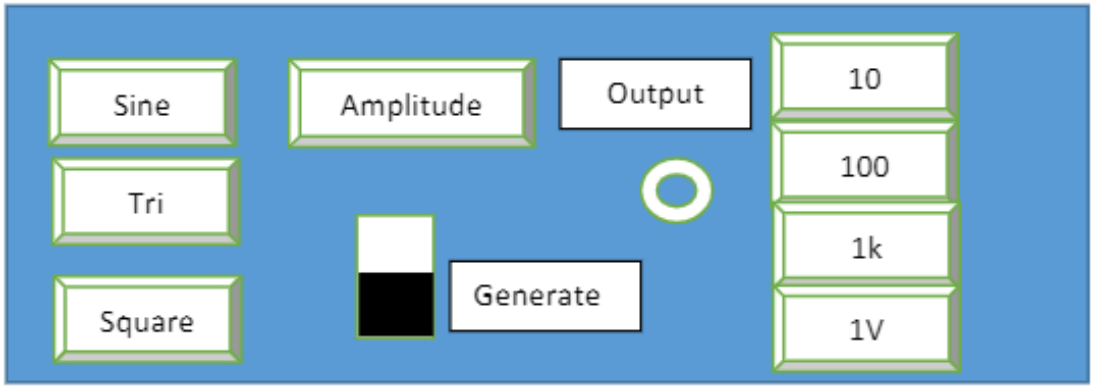
CS/EEE/ECE/INSTR F241
MICROPROCESSORS, PROGRAMMING AND INTERFACING

BITS PILANI K K BIRLA GOA CAMPUS

Table of Contents

User Requirements & Technical Specifications

This system is used to generate a Sine/Triangular/Square waveform of Frequencies ranging from 10 Hz to 99KHz. Voltage is between 0-10V.



Technical Specifications:

- On system power-up the user has to configure the desired type of waveform (square/triangle/square), frequency and amplitude.
- To generate a Square Waveform of Frequency 9.35KHz the user has to press square key, followed by 1K Key – 9 Times, 100 Key – 3 Times 10 Key – 5 Times.
- To select the Amplitude the user will have to press Amplitude key and then press the 1V key “n” number of times where “n” is the peak-to-peak amplitude of the waveform to be generated. (only integer values of output voltages need to be generated)
- When generate switch should be turned on and then the frequency generation is enabled i.e., the square waveform of that frequency will be generated.
- When frequency generation is enabled, if the user wants to change the waveform into another type for e.g., sine he just has to press sine.
- When a signal of different type as well as different amplitude/frequency has to be generated, the user will have to turn-off the generate switch and then configure the function generator as mentioned above.

Assumptions & Justifications

Assumptions

1. The interval between each user key-press is sufficient to complete all operations associated with it.
2. The user can only increment the value of the required Amplitude. In order to decrease it, they will have to restart the system, or complete this generation sequence (inputting frequency, mode, amplitude), and toggling Generate switch ON and OFF, to achieve the same effect.
3. The user inputs the frequency efficiently, using minimum number of frequency key presses, representing real life usage. However, they are free to keep changing Waveform type whenever they want.
4. Inferring from the technical specifications, there exists a specific sequence the user has to follow to use the frequency generator, wherein they cannot progress to next stage without completing the previous one:
    a. User presses any of the mode buttons at least once. They can continue pressing mode buttons throughout the rest of the operation sequence, but they must start with a mode button press.
    b. User presses the required frequency buttons to get required frequency
    c. User presses the AMP key, signaling that required frequency has been finalized, and that they're about to input the required amplitude.
    d. User presses the 1V key required number of times
    e. User toggles the Generate Switch ON
    f. If required, the User is free to change wave mode during this output stage
    g. User toggles the Generate Switch OFF, which returns the system back to stage 'a'.

Justifications

1. Due to the presence of *1K*, *100* as well as *10* Keys, and due to the max-generated-frequency limit of 99KHz, the system only allows a maximum of 99 *1K-key* presses, 9 *100-key* presses, and 9 *10-key* presses by the user. This enables the user to decrease the frequency count by going above the max key count, which resets the key count to zero.

Components used with Justification wherever required

- 8086 x1
- 8284 x1
- Digital-to-Analog Converter x1
  - **DAC0830:** Used to convert the discrete signal inputs taken from 8086 (via 8255) into a continuous current waveform.
  - Covers the required output voltage range of 0-10V.
  - Also provides a Feedback Resistor pin R_{Fb}, which can be used with an external Operational Amplifier to convert the current waveform into a voltage waveform.
  - Operation Equations given below: (further details in manual and design pdf)

$$I_{OUT1} = \frac{V_{REF}}{15k\Omega} \times \frac{Digital\ Input}{255}$$

$$I_{OUT2} = \frac{V_{REF}}{15k\Omega} \times \frac{255 - Digital\ Input}{255}$$

  - Detailed manual attached

- Operational Amplifier x1
  - **LM741:** An Integrated Circuit that can amplify weak electric signals.
  - Can be used in conjunction with a DAC to convert the DAC's current output into a voltage output.
  - Here it is shunted with DAC's feedback resistor pin Rfb, to generate the required analog voltage signal, from the DAC's analog current signal output.
  - Operation Equations (in above context) given below: (further details in manual and design pdf)

$$V_{OUT} = V_{REF} \times \frac{Digital\ Input}{255}$$

  - Detailed manual attached

- 4x3 Matrix Keypad x1
  - **AK304:** Used to accept inputs from the user.
  - Chosen considering operational conditions and the requirement of 8 momentary action mechanical switches/buttons.
  - 8 buttons of the keypad are used (for selecting the mode, frequency and amplitude for wave generation), while the other 4 remain unmapped.
  - Detailed manual attached.
- 8255 x1
  - Used to interface DAC and AK304 with 8086.

- 8254A x1
  - Used for generating an Interrupt timer (for 8259).
- 8259 x1
  - Generates Interrupts from 4 sources:
    - Generate switch toggled ON (instructs 8086 to begin output phase)
    - Timer from 8254A (instructs 8086 to push next discrete value to DAC via 8255)
    - Mode button pressed mid-output (instructs 8086 to change output table)
    - Generate switch toggled OFF (instructs 8086 to end output phase)

- Debouncing IC x1
  - **MAX6818:** Octal switch debouncer that provides clean interfacing of mechanical switches to digital systems.
  - Used to debounce the seven AK304 signals before connecting to 8255 to reduce reliance on software methods of debouncing, as well as to enable the usage of the mode buttons directly as a trigger for an interrupt (since the output of this debouncing circuit can be directly connected to digital systems).
  - Detailed manual attached

- SPDT Toggle Switch x1
  - **21236N:** An SPDT Toggle-switch with latching action is used for generating the wave.
  - Chosen due to its simple debouncing circuitry requirements and to function as the desired toggle-able Generate switch.
  - Detailed manual attached

- 2716 x4
  - The smallest available ROM chip is 2K. We need to have an even and odd bank.
  - Further, we require two pairs of the banks, one at the starting addresses 0000h onward (to store the IVT), and one at the ending addresses FF000h onward (to store the jump-to-start statement at reset address FFFF0h)
- 6116 x4
  - Smallest available RAM chip is 2K and we require odd and even banks. RAM is used for temporary storage of data and stack.
  - Two such pairs of banks are used to provide enough temporary storage for smooth operation.
- LS138 x2
  - Decoder used for Memory and I/O decoding.
- LS373, LS244, LS 245, and required gates

Address Map

Memory Map

ROM_1	:	$00000_H – 00FFF_H$
RAM_1	:	$01000_H – 01FFF_H$
RAM_2	:	$02000_H – 02FFF_H$
ROM_2	:	$FF000_H – FFFFF_H$

I/O Map

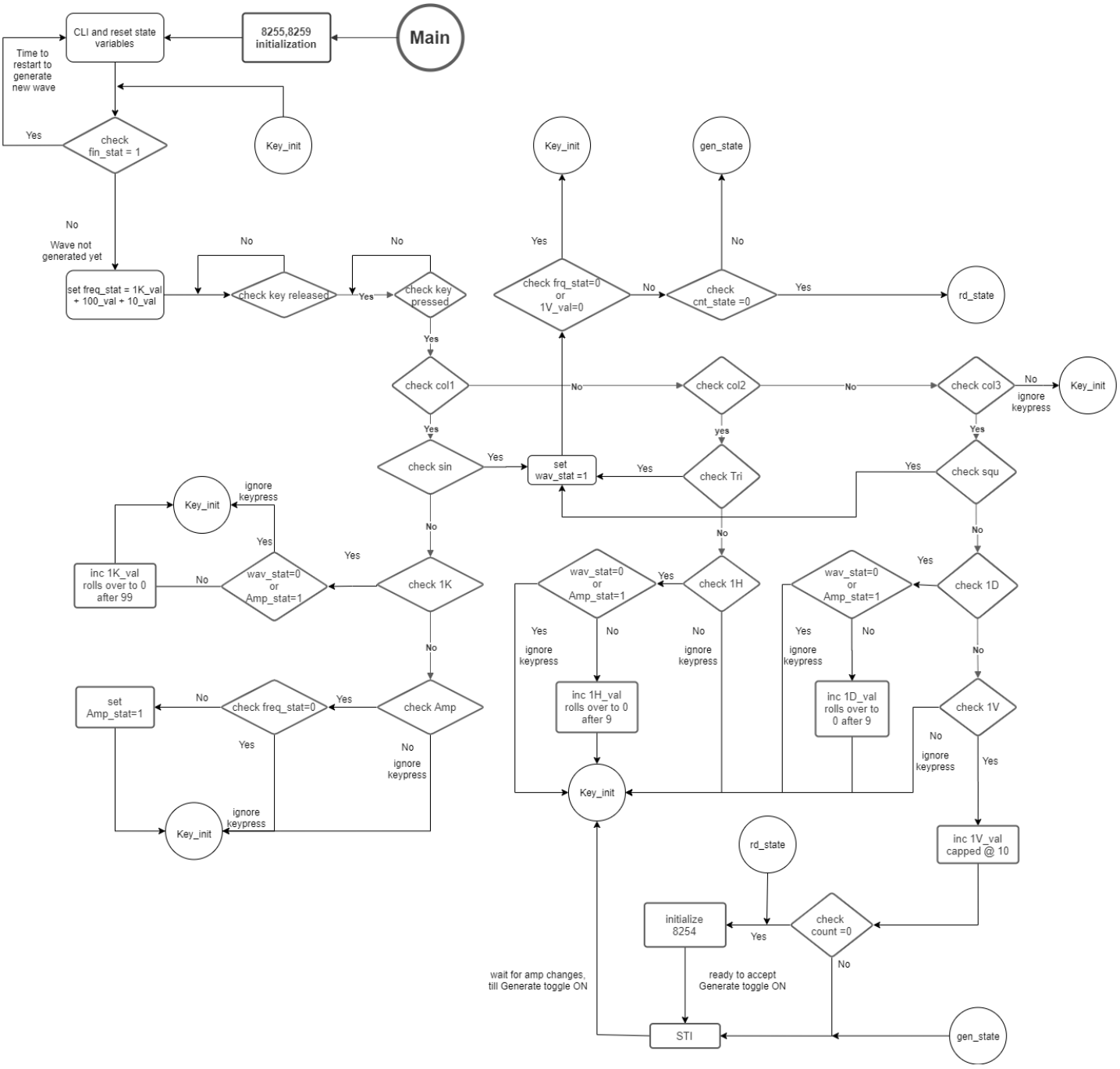
8255	:	$80_H – 86_H$
8254	:	$90_H – 96_H$
8259	:	$A0_H – A2_H$

Design

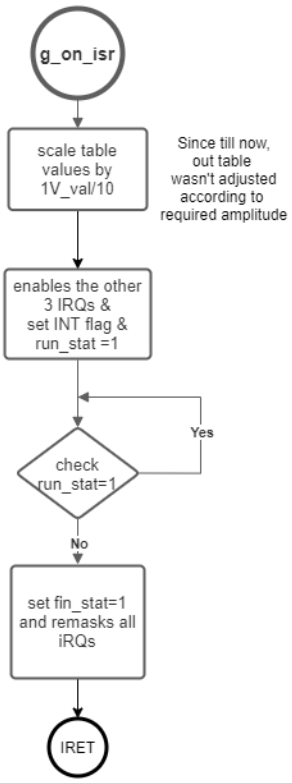
Complete design is shown with proper labelling in attached file *freq_gen_g9_pinout.pdf*

Flowchart

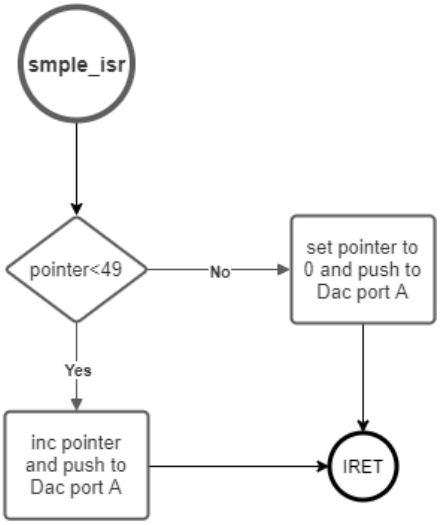
Main Program



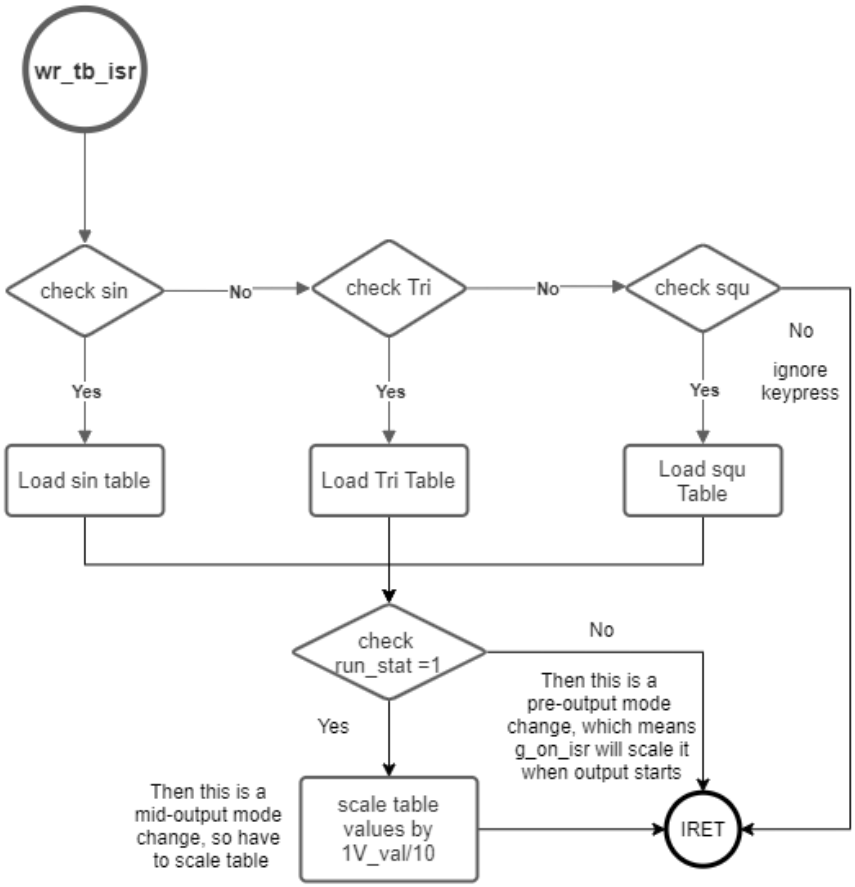
ISR of INT 43_H: Generate Toggled ON



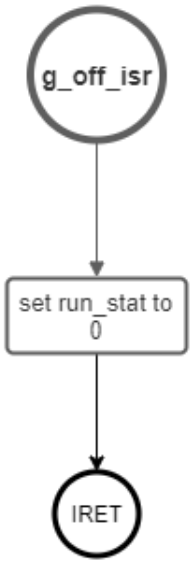
ISR of INT 42_H: Push Data to DAC



ISR of INT 41_H: Wave Mode Change



ISR of INT 40_H: Generate Toggled OFF



Variations in Proteus Implementation with Justification

- As 8259 does not work in proteus, Interrupts are all replaced with equivalent polling programming logic.
- ROM in only 00000 – as proteus allows to change reset address.
- Using 8253 – as 8254 not available in Proteus.
- Clock is at 2 MHz as the clock generated for 8086 requires a long rise and fall time of clock. Thus, the 8253 DAC Push clock will also function with the same clock.
- 2732 is used as 2716 – not available in Proteus.
- Using a gate-based circuit for memory – does the same as LS 138 here.
- 8259 not there – justification is as per the first point.

Firmware

System is implemented using the fully annotated attached emu8086 asm file *freq_gen_g9.asm* and the attached emu8086 bin file *freq_gen_g9.bin*

List of Attachments

1. Complete Hardware Real World Design - *freq_gen_g9_pinout.pdf*
2. Manuals - *DAC0830.pdf*
   - *LM741.pdf*
   - *MAX6818.pdf*
   - *SN74LVC1G04.pdf*
   - *21236N.pdf*
   - *AK304.pdf*
3. Proteus File - *freq_gen_g9.dsn*
4. emu8086 asm file - *freq_gen_g9.asm*
5. Binary file after assembly - *freq_gen_g9.bin*