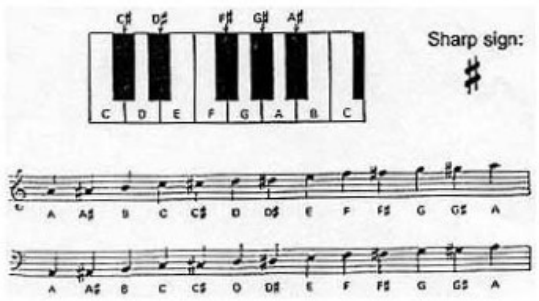
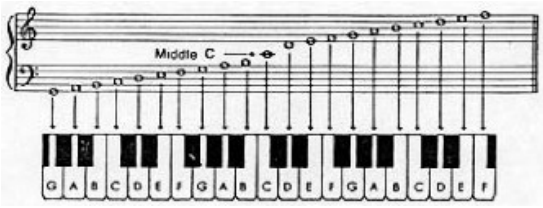
Project 2: Wave-Table Synthesizer

Synthesizing Music

A musical note is specified by three parameters: frequency, duration, and loudness. We will assume the middle A note is 440Hz (usual calibration for piano). Notes in the scale of C begin with middle C, D, E, F, G, A, B, and C where the final C is one octave higher than the middle-C are scaled with a doubling, halving in frequency for each octave increase (next scale), decrease (previous scale) respectively. Notations for scale and octave are shown below and frequency data for each note is also given below.





	OCTAVE NUMBER								
	0	1	2	3	4	5	6	7	8
C	16.3516	32.7032	65.4064	130.813	261.626	523.251	1046.50	2093.00	4186.01
C#	17.3239	34.6478	69.2957	138.591	277.183	554.365	1108.73	2217.46	4434.92
D	18.3540	36.7081	73.4162	146.832	293.665	587.330	1174.66	2349.32	4698.64
D#	19.4454	38.8909	77.7817	155.563	311.127	622.254	1244.51	2489.02	4978.03
E	20.6017	41.2034	82.4069	164.814	329.628	659.255	1318.51	2637.02	5274.04
F	21.8268	43.6536	87.3071	174.614	349.228	698.456	1396.91	2793.83	5587.65
F#	23.1247	46.2493	92.4986	184.997	369.994	739.989	1479.98	2959.96	5919.91
G	24.4997	48.9994	97.9989	195.998	391.995	783.991	1567.98	3135.96	6271.93
G#	25.9565	51.9131	103.826	207.652	415.305	830.609	1661.22	3322.44	6644.88
A	27.5000	55.0000	110.000	220.000	440.000	880.000	1760.00	3520.00	7040.00
A#	29.1352	58.2705	116.541	233.082	466.164	932.328	1864.66	3729.31	7458.62
B	30.8671	61.7354	123.471	246.942	493.883	987.767	1975.53	3951.07	7902.13

Memory Layout

Parameter definitions and a memory layout is provided in the wavesyn.dat file. This file is listed in the texbook and the latest version may be downloaded from the EE592 website.

Algorithm

We will generate notes using the following steps.

1) The desired frequency will dictate Δ (offset) used in sinusoid synthesis

2) We will assume a duration of one second for all notes (in real music this is obviously not true)

3) An additional scale factor, loudness, will be used to further scale the synthesized tone for periods of low or high intensity

```
INITIALIZE
       Set NEW_NOTE_FLAG, A_FLAG; Clear DS_FLAG, R_FLAG
       Set ADSR_COUNTER = A_DURATION

-------------------------------------------------------------------------------------------------------------------
MAIN (once per sample period)
       If NEW_NOTE_FLAG
              GET_NOTE_DATA
              Clear NEW_NOTE_FLAG
       End
       SYNTHESIZE_NOTE
       CHECK_ADSR_COUNTER
       CHECK_NOTE_COUNTER
       Goto MAIN

-------------------------------------------------------------------------------------------------------------------
GET_NOTE_DATA
       get NOTE_DELTA (integer and fractional portions)
       get NOTE_DURATION
       get NOTE_LOUDNESS
       set NOTE_COUNTER = NOTE_DURATION

-------------------------------------------------------------------------------------------------------------------
SYNTHESIZE_NOTE
       Based on DELTA, GET_SINUSOID_VALUE
       COMPUTE_ADSR_WEIGHT
       TMP = (SINUSOID_VALUE)(ADSR_WEIGHT)
       OUTPUT = (TMP)(NOTE_LOUDNESS)

-------------------------------------------------------------------------------------------------------------------
CHECK_ADSR_COUNTER
       Decrement ADSR_COUNTER (tracks progress through portions of envelope)
       If ADSR_DURATION_COUNTER = 0 (new ADSR portion)
              If A_FLAG
                     Clear A_FLAG, set DS_FLAG, ADSR_COUNTER = DS_DURATION
              Elseif DS_FLAG
                     Clear DS_FLAG, set R_FLAG, ADSR_COUNTER = R_DURATION
              Else (must be R_FLAG)
                     Clear R_FLAG, set A_FLAG, ADSR_COUNTER = A_DURATION
              End
       End

-------------------------------------------------------------------------------------------------------------------
CHECK_NOTE_COUNTER
       Decrement NOTE_DURATION_COUNTER (tracks progress through note)
       If NOTE_COUNTER = 0
              Set NEW_NOTE_FLAG
       End

-------------------------------------------------------------------------------------------------------------------
GET_SINUSOID_VALUE (EE442)
       Generate sinusoid value using the linear interpolation method

-------------------------------------------------------------------------------------------------------------------
COMPUTE_ADSR_WEIGHT
       If A_FLAG (in the attack portion of the ADSR envelope)
              Get filter parameters, A_HAT and G for attack phase
       Elseif DS_FLAG (in the decay/sustain portion of the ADSR envelope)
              Get filter parameters, A_HAT and G for decay/sustain phase
       Else (in the release portion of the ADSR envelope)
              Get filter parameters, A_HAT and G for release phase
       End;
       ADSR_WEIGHT = (A_HAT)(G)+(1-G)(ADSR_WEIGHT)
```