TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

2018-06-12 (72 회차)

강사 - Innova Lee(이상훈) gcccompil3r@gmail.com 학생 - 정유경

```
1. FFT 개념
                                                                           GLfloat n range = 100.0f;
                                                                                                                                                       while(cnt++ < 36) {
2. cos dft
                                                                                                                                            printf("%.1fcos(%f * %.8f) = %f\n", amplitude, and vel,
3. sin fft
                                                                           if(h == 0)
                                                                                                                                                       t, amplitude * cos(ang vel * t));
                                                                                h = 1:
                                                                                                                                                                 t += step:
                                                                           alViewport(0, 0, w, h):
                                                                                                                                            }
                                                                           glMatrixMode(GL PROIECTION):
cos dft.c
                                                                           glLoadIdentity();
                                                                                                                                            void sin sim(float amplitude, float ang vel, float period){
                                                                                                                                                       int cnt = 0:
#include <stdio.h>
                                                                           if(w \le h)
                                                                                                                                                       float step, t = 0.0;
#include <stdlib.h>
                                                                                glOrtho(-n range, n range, -n range * h / w, n range * h /
#include <string.h>
                                                                     w, -n range, n range);
                                                                                                                                                      t = step = get step(SLICE, period);
#include <time.h>
#include <math.h>
                                                                                glOrtho(-n range * w / h, n range * w / h, -n range.
                                                                                                                                                       while(cnt++ < 36) {
                                                                      n range, -n range, n range);
                                                                                                                                            printf("\%.1fsin(\%f * \%.8f) = \%f\n", amplitude, ang vel,
                                                                                                                                                                           t, amplitude * sin(ang vel * t));
#include <GL/glut.h>
                                                                           glMatrixMode(GL MODELVIEW);
                                                                                                                                                                 t += step:
#define SLICE
                                                                           glLoadIdentity();
void draw omega sin(void);
void draw spectrum(void);
                                                                      void keyboard(unsigned char key, int x, int y){
                                                                                                                                            void draw spectrum(void){
                                                                                                                                                       float x = 0, x^2 = 0, y^2, cx, cy;
                                                                                switch(kev)
float common angles[5] = \{15.0, 30.0, 45.0, 60.0, 75.0\}:
                                                                                                                                                       float t. step = 0.0:
float freq table[5] = \{1000.0, 2400.0, 5000.0, 24000.0, 77000.0\};
                                                                                                                                                       int i. i. cnt = 0. cache = 0:
                                                                                           case 27:
                                                                                                                                                       float period, freq = 100.0;
                                                                                                     exit(0);
float theta = 0.0:
                                                                                                     break;
                                                                                                                                                       float res real[32] = \{0\};
                                                                                                                                                       float res image[32] = \{0\};
                                                                                                                                                       float y[32] = \{0\};
typedef struct complex{
                                                                     }
          float cosx[32][32];
                                                                                                                                                       c \exp = \{0\}:
          float isinx[32][32]:
                                                                      void set rand amplitude(float *amp){
                                                                                *amp = rand() % 3 + 3:}
} c;
                                                                                                                                                       calc period(&freq. &period):
                                                                                                                                                       step = get step(SLICE, period);
void display(void){
                                                                      void set angle with common angles(float *angle){
          glClearColor(0.0, 0.0, 0.0, 1.0);
                                                                                *angle = common angles[rand() % 5];}
                                                                                                                                                       for(i = 0: i < SLICE: i++)
          glClear(GL COLOR BUFFER BIT | GL DEPTH BUFFER BIT);
                                                                      void angle2radian(float *angle, float *radian){
          glLoadIdentity();
                                                                                                                                                                 for(j = 0; j < SLICE; j++)
                                                                                *radian = *angle * M PI / 180.0;}
         //gluLookAt(0.0, 0.0, 3.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
                                                                                                                                                       \exp.cosx[i][j] = cos(-(2 * M PI * j * i) / SLICE);
                                                                                                                                                       \exp.isinx[i][i] = sin(-(2 * M PI * i * i) / SLICE):
                                                                      void radian2angle(float *angle, float *radian){
          glColor3f(1, 0, 0);
                                                                                *angle = *radian * 180.0 / M PI;}
                                                                                                                                                       printf("exp.cosx[%d][%d] = %f\n", i, i, exp.cosx[i][i]);
                                                                                                                                                       //printf("exp.isinx[%d][%d] = %f\n", i, j, exp.isinx[i][j]);
          glBegin(GL LINE LOOP);
                                                                      void set rand frequency(float *freq){
         glVertex3f(100.0, 0.0, 0.0);
                                                                                *freg = freg table[rand() % 5];}
    glVertex3f(-100.0, 0.0, 0.0);
    glEnd();
                                                                      void calc period(float *freq, float *period){
                                                                                                                                                       if(t > period)
                                                                                *period = 1 / (*freq):
                                                                                                                                                                 t = 0.0:
    glColor3f(0.0, 1.0, 0.0);
                                                                      void calc angular velocity(float *freq, float *ang vel){
                                                                                                                                                      i = 0;
                                                                                *ang vel = 2 * M PI * (*freg); }
    glBegin(GL LINE LOOP);
                                                                                                                                                      t = 0.0:
    alVertex3f(0.0, 100.0, 0.0);
                                                                                                                                                       for(; i < SLICE; t += step)
                                                                      float get step(float slice, float period){
    glVertex3f(0.0, -100.0, 0.0);
    alEnd():
                                                                                return period / slice: }
                                                                                                                                                                 //if(t > 3 * period)
                                                                                                                                                                 if(t > period)
         //draw_omega_sin();
                                                                      void cos sim(float amplitude, float ang vel, float period){
          draw spectrum();
                                                                                int cnt = 0;
                                                                                                                                                                           break;
          glutSwapBuffers();
                                                                                float step, t = 0.0;
                                                                                                                                                                           t = 0.0;
}
                                                                                t = step = get step(SLICE, period);
void reshape(int w. int h){
                                                                                                                                                                 v[i] = 10 * cos(200 * M PI * t):
```

```
printf("y[\%d] = \%f\n", i++, y[i]);
                                                                                                        glEnd();
                     //printf("exp.cosx[%d] = %f\n", i, exp.cosx[i]);
                     //printf("exp.isinx[%d] = %f\n", i, exp.isinx[i]);
                     //printf("res real[%d] = %f\n", i, res real[i]);
                                                                                              cache = 1:
                     //printf("res real = %f\n", res real);
                                                                                              cx = t:
                     //printf("res image = %f\n", res image);
                                                                                              cy = y2;
                                                                                              cnt++:
          for(i = 0; i < SLICE; i++)
                                                                        #endif
                     for(j = 0; j < SLICE; j++)
                               res_real[i] += y[j] * exp.cosx[i][j];
res_image[i] += y[j] * exp.isinx[i][j];
                                                                        void draw omega sin(void){
                                                                                   float amp, angle, period, freq, rad, omega, t, step = 0.0;
                                                                                   float radius = 3.0:
                               //printf("res real[%d] = %f\n", i,
res real[i]);
                                                                                   float x = 0, x^2 = 0, y^2, cx, cy;
                                printf("res image[%d] = %f\n", i,
                                                                                   float tmp:
                                                                                   int cnt = 0, cache = 0;
res_image[i]);
                                                                                   srand(time(NULL));
          //printf("OK");
                                                                                   amp = 10;
                                                                                   angle = 45.0:
          for(i = 0; i < SLICE; i++)
                                                                                   frea = 100.0:
                     glBegin(GL POINTS);
                                                                                   angle2radian(&angle, &rad);
                     glVertex2f(i * 10, res real[i] * 3);
                                                                                   calc period(&freg, &period);
                     //glVertex2f(i * 10, res_image[i] * 3);
                                                                                   calc angular velocity(&freg, &omega);
                                                                                   t = step = get step(SLICE, period);
                     glEnd();
                     glBegin(GL LINE STRIP);
                                                                                   printf("step = \%f\n". step):
                     glVertex2f(i * 10, res real[i] * 3);
                     alVertex2f(i * 10, 0):
                                                                                   if(t > period)
                     //glVertex2f(i * 10, res image[i] * 3);
                                                                                             t = 0.0;
                     //glVertex2f(i * 10, 0);
                                                                                   //qlLineStipple(1, 0xFFEE);
                     glEnd();
                                                                                   //glEnable(GL LINE STIPPLE);
#if 0
                                                                                   //glBegin(GL POINTS);
          glBegin(GL LINE STRIP);
                                                                                   for(; ; t += step)
          for(i = 0; i < SLICE; i++)
                                                                                              if(t > period)
                                                                                              //if(t > 3 * period)
                     glVertex2f(j * 10, res real[j] * 2);
                     glVertex2f(j * 10, 0);
                                                                                                        break;
                                                                                                        t = 0.0:
          glEnd();
#endif
                                                                                             y2 = amp * sin(omega * t);
#if 0
                     //if(cache && !(cnt % 16))
                                                                                              if(cache && !(cnt % 16))
                     if(cache)
                                                                                                        glBegin(GL POINTS);
                                alBeain(GL POINTS):
                                                                                                        gVertex2f(cx * 6000, cy * 6);
                                glVertex2f(cx * 4000, cy * 6);
                                                                                                        glVertex2f(t * 6000, y2 * 6);
                                glVertex2f(t * 4000, y2 * 6);
                                                                                                        glEnd();
                                glEnd();
                                                                                                        glBegin(GL LINE STRIP);
                                                                                                        //glVertex2f(cx * 4000, cy * 2);
                                glBegin(GL LINE STRIP);
                                gIVertex2f(t * 4000, y2 * 6);
                                                                                                        //glVertex2f(cx * 4000, 0);
                                alVertex2f(t * 4000. 0):
                                                                                                        alVertex2f(t * 6000, v2 * 6):
```

```
glVertex2f(t * 6000, 0);
                              glEnd();
                    }
                    cache = 1;
                    cx = t:
                    cy = y2;
                    cnt++;
         //glEnd();
         //glDisable(GL LINE STIPPLE);
int main(int argc, char **argv){
          float amplitude, angle, period, frequency, radian,
angular velocity;
         float step = 0.0;
          glutInit(&argc, argv);
          glutInitDisplayMode(GLUT DOUBLE);
          glutInitWindowSize(1200, 800);
          alutInitWindowPosition(0, 0):
          alutCreateWindow("Digital Signal Processing"):
          glutDisplayFunc(display);
          glutReshapeFunc(reshape);
         glutMainLoop();
          return 0;
}
```

```
sin fft.c
                                                                     void reshape(int w, int h)
#include <stdio.h>
                                                                          GLfloat n range = 100.0f;
#include <stdlib.h>
#include <string.h>
                                                                          if(h == 0)
#include <time.h>
                                                                               h = 1:
#include <math.h>
                                                                          glViewport(0, 0, w, h);
                                                                          glMatrixMode(GL PROJECTION);
#include <GL/glut.h>
                                                                          glLoadIdentity();
#define SLICE
                              (SLICE >> 1)
#define HALF SLICE
                                                                          if(w \le h)
                                                                     glOrtho(-n range, n range, -n range * h / w, n range * h / w.
int alob = 4:
                                                                     -n range, n range):
int count = 0:
                                                                          else
                                                                     glOrtho(-n range * w / h, n range * w / h, -n range, n range,
                                                                     -n range, n range);
#define TWID FACTOR (SLICE >> 1) + 1
                                                                          glMatrixMode(GL_MODELVIEW);
typedef struct complex
                                                                          glLoadIdentity();
     double re;
     double im:
} c;
                                                                     void keyboard(unsigned char key, int x, int y){
                                                                               switch(key)
void draw omega sin(void);
void draw spectrum(void);
                                                                                         case 27:
                                                                                                    exit(0);
float common angles[5] = \{15.0, 30.0, 45.0, 60.0, 75.0\};
                                                                                                   break;
float freq table[5] = \{1000.0, 2400.0, 5000.0, 24000.0, 77000.0\};
float theta = 0.0:
                                                                     void set rand amplitude(float *amp){
void display(void)
                                                                               *amp = rand() % 3 + 3;}
          glClearColor(0.0, 0.0, 0.0, 1.0);
                                                                     void set angle with common angles(float *angle){
          glClear(GL COLOR BUFFER BIT | GL DEPTH BUFFER BIT);
                                                                               *angle = common angles[rand() % 5];}
          glLoadIdentity();
                                                                     void angle2radian(float *angle, float *radian){
         //qluLookAt(0.0, 0.0, 3.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
                                                                               *radian = *angle * M PI / 180.0:}
                                                                     void radian2angle(float *angle, float *radian){
          qlColor3f(1, 0, 0);
                                                                               *angle = *radian * 180.0 / M PI;}
          glBegin(GL LINE LOOP);
          glVertex3f(100.0, 0.0, 0.0);
                                                                     void set rand frequency(float *freq){
                                                                               *freq = freq table[rand() % 5];}
    glVertex3f(-100.0, 0.0, 0.0);
    glEnd();
                                                                     void calc period(float *freq, float *period){
    glColor3f(0.0, 1.0, 0.0);
                                                                               *period = 1 / (*freq);}
                                                                     void calc angular velocity(float *freq, float *ang vel){
    alBeain(GL LINE LOOP):
    alVertex3f(\overline{0}.0, \overline{10}0.0, 0.0);
                                                                               *ang vel = 2 * M PI * (*freq);}
    alVertex3f(0.0. -100.0. 0.0):
                                                                     float get step(float slice, float period){
    glEnd();
                                                                               return period / slice;}
         //draw omega sin();
          draw spectrum();
                                                                     void cos_sim(float amplitude, float ang_vel, float period){
          glutSwapBuffers();
                                                                               int cnt = 0;
                                                                               float step, t = 0.0;
}
```

```
t = step = get step(SLICE, period);
          while(cnt++ < 36) {
          printf("%.1fcos(%f * %.8f) = %\\n", amplitude. and vel.
                               t, amplitude * cos(ang vel * t));
}
void sin sim(float amplitude, float ang vel, float period){
          int cnt = 0:
          float step, t = 0.0;
          t = step = get step(SLICE, period):
          while(cnt++ < 36)
          printf("\%.1fsin(\%f * \%.8f) = \%f\n", amplitude, and vel,
                     t, amplitude * sin(ang vel * t));
                     t += step;
}
void draw spectrum(void){
          float t, step = 0.0;
          float period, freq = 100.0;
          c y[8] = \{0\};
     int ix:
     int iu:
     int iy;
     int i:
     double x[8] = \{0\};
          int tst;
     double temp re;
     double temp im;
     int iheight;
     int istart:
     int j;
     double twid re;
     double dv0[5] = \{0\};
          double twid im:
          double dv1[\overline{5}] = \{0\};
          int ihi:
          calc period(&freg, &period);
          step = get step(SLICE, period);
          if(t > period)
                     t = 0.0:
          i = 0;
          t = 0.0;
          for(; i < SLICE; t += step)
                     //if(t > 3 * period)
```

```
if(t > period)
                                                                                                                                                                                       i += ix:
                                                                              for (i = 0; i \le 7; i += 2) {
                           break:
                                                                                         temp re = y[i + 1].re;
                          t = 0.0:
                                                                                         temp im = y[i + 1].im;
                                                                                                                                                                            istart++:
                                                                                         y[i + 1].re = y[i].re - y[i + 1].re;
                                                                                         y[i + 1].im = y[i].im - y[i + 1].im;
                x[i] = 10 * sin(200 * M PI * t):
                                                                                         y[i].re += temp re;
                                                                                                                                                                  iu /= 2:
                printf("x[%d] = %f\n", \bar{i}++, x[i]);
                                                                                         y[i].im += temp im;
                                                                                                                                                                  iy = ix;
                //printf("exp.cosx[%d] = %f\n", i, exp.cosx[i]);
                                                                                                                                                                  ix <<= 1;
                //printf("exp.isinx[%d] = %f\n", i, exp.isinx[i]);
                                                                                         printf("y[%d].re = %lf\t", i, y[i].re);
                                                                                                                                                                  iheight -= iy;
                //printf("res real[%d] = %f\n", i, res real[i]);
                                                                                         printf("v[\%d].im = \%lf\n", i, v[i].im);
                //printf("res_real = %f\n", res_real);
                                                                                         printf("y[\%d].re = \%lf\t", i+1, y[i+1].re);
                                                                                                                                                                  if(ju > 0)
                                                                                         printf("v[%d].im = %lf\n". i+1. v[i+1].im):
                //printf("res image = %f\n", res image);
                                                                                                                                                                            count++:
                                                                                                                                            printf("\nFinished N-%d Butterfly\nNow Starting N-%d Butterfly",
t = 0:
                                                                              iy = 2;
                                                                                                                                            alob. alob *= 2):
     step = 2 * M PI / SLICE;
                                                                              ix = 4;
                                                                                                                                                                  else
                                                                              iu = 2:
for(i = 0; i < 5; i++)
                                                                              iheight = 5;
                                                                                                                                            printf("\nFinished N-%d Butterfly\n", glob);
                                                                              while (ju > 0) {
     dv0[i] = cos(t);
                                                                                         // 0 ~ 4
     dv1[i] = -sin(t):
                                                                                         printf("\nN-%d Butterflv(처음은 짝수 오더)\n".
                                                                                                                                                       //printf("OK");
     t += step:
                                                                   alob):
}
                                                                                         for (i = 0; i < iheight; i += ix) {
                                                                                                                                                       for(i = 0; i < SLICE; i++)
                                                                                         temp re = y[i + iy].re;
                                                                                         temp im = y[i + iy].im;
     t = 0:
                                                                                                                                                                  glBegin(GL LINE STRIP);
                                                                                         y[i + iy].re = y[i].re - temp_re;
                                                                                         y[i + iy].im = y[i].im - temp im;
     ix = 0;
                                                                                                                                                                  if(y[i].re == 0 \&\& y[i].im == 0)
     iu = 0:
                                                                                         y[i].re += temp re;
                                                                                                                                                                            continue:
                                                                                         v[i].im += temp im:
     iv = 0:
                                                                                         printf("y[\%d].re = \%lf\t", i, y[i].re);
                                                                                                                                                                  glVertex2f(i * 10. v[i].re / HALF SLICE):
     printf("Before Reverse Order\n"):
     for (i = 0; i < 7; i++) {
                                                                                         printf("v[\%d].im = \%lf\n". i. v[i].im):
                                                                                                                                                                  alVertex2f(i * 10. 0):
                y[iy].re = x[ix];
                                                                                         printf("y[\%d].re = \%lf\t", i+iy, y[i+iy].re);
                y[iy].im = 0.0;
                                                                                         printf("y[\%d].im = \%lf\n", i+iy, y[i+iy].im);
                                                                                                                                                                  alEnd():
                printf("y[%d].re = %lf\t", iy, y[iy].re);
                                                                                         printf("\n");
                                                                                                                                                                  glBegin(GL LINE STRIP);
                printf("y[%d].im = %lf\n", iy, y[iy].im);
                                                                                         istart = 1;
                                                                                                                                                                  glVertex2f(i * 10, y[i].im / HALF SLICE);
                                                                              printf("\nN-%d Butterflv(처음은 홀수 오더)\n". glob):
                iv = 8:
                                                                                                                                                                  alVertex2f(i * 10. 0):
                tst = 1:
                                                                                         for (i = iu; i < 4; i += iu) {
                                                                              printf("twid re = dv0 = cos(2 * pi * f * t / fftN 의절반)\n");
                while (tst) {
                                                                                                                                                                  glEnd();
                                                                              printf("twid_im = dv1 = -sin(2 * pi * f * t / fftN 의절반)\n");
                                                                                                                                                       }
                           iy >>= 1;
                          iu ^= iv:
                                                                                                   twid re = dv0[i]:
                           tst = ((ju \& iy) == 0);
                                                                                                   twid im = dv1[i]:
                                                                                                                                            #if 0
                                                                                                   i = istart:
                                                                                                                                                                  //if(cache && !(cnt % 16))
                                                                                                   ihi = istart + iheight:
                                                                                                                                                                  if(cache)
                                                                                                   while (i < ihi) {
                iy = ju;
                                                                              temp re = twid re * y[i + iy].re - twid im * y[i + iy].im;
                                                                                                                                                                            glBegin(GL POINTS);
                ix++;
                                                                              temp im = twid re * y[i + iy].im + twid im * y[i + iy].re;
                                                                                                                                                                            gVertex2f(cx * 4000, cy * 6);
     y[iy].re = x[ix]:
                                                                              y[i + iy].re = y[i].re - temp re:
                                                                                                                                                                            g|Vertex2f(t * 4000, y2 * 6);
     y[iy].im = 0.0;
                                                                              y[i + iy].im = y[i].im - temp im;
                                                                                                                                                                            alEnd():
                                                                              y[i].re += temp re;
     printf("\nAfter Reverse Order\n");
                                                                              y[i].im += temp im;
                                                                                                                                                                            glBegin(GL LINE STRIP);
     for(i = 0; i < SLICE; i++)
                                                                                                                                                                            glVertex2f(t * 4000, y2 * 6);
                                                                              printf("y[%d].re = %lf\t", i, y[i].re / HALF_SLICE);
                                                                                                                                                                            glVertex2f(t * 4000, 0);
                printf("y[\%d].re = \%lf\t", i, y[i].re);
                                                                              printf("y[%d].im = %lf\n", i, y[i].im / HALF SLICE);
                                                                                                                                                                            glEnd();
                                                                              printf("v[\%d].re = \%lf\t", i+iv, v[i+iv].re / HALF SLICE);
                printf("v[%d].im = %lf\n", i, v[i].im);
                                                                              printf("y[\%d].im = \%lf\n", i+iy, y[i+iy].im / HALF SLICE);
     printf("\nN-2 First Butterflv\n"):
                                                                                                                                                                  cache = 1:
```

```
cx = t;
                    cy = y2;
                                                                                          cache = 1;
                    cnt++;
                                                                                          cx = t;
                                                                                          cy = y2;
#endif
                                                                                         cnt++;
                                                                               //glEnd();
void draw omega sin(void)
                                                                               //glDisable(GL LINE STIPPLE);
          float amp, angle, period, freq, rad, omega, t, step = 0.0;
         float radius = 3.0;
                                                                     int main(int argc, char **argv)
         float x = 0, x^2 = 0, y^2, cx, cy;
                                                                                float amplitude, angle, period, frequency, radian,
          float tmp;
         int cnt = 0, cache = 0:
                                                                     angular velocity;
                                                                               float step = 0.0;
         srand(time(NULL));
                                                                                glutInit(&argc, argv);
          amp = 10;
                                                                                glutInitDisplayMode(GLUT_DOUBLE);
         angle = 45.0;
                                                                               glutInitWindowSize(1200, 800);
                                                                               glutInitWindowPosition(0, 0);
         freq = 100.0;
                                                                                glutCreateWindow("Digital Signal Processing");
          angle2radian(&angle, &rad);
          calc period(&freq, &period);
                                                                                glutDisplayFunc(display);
          calc angular velocity(&freq, &omega);
                                                                               glutReshapeFunc(reshape);
         t = step = get_step(SLICE, period);
                                                                               glutMainLoop();
          printf("step = \%f\n", step);
                                                                               return 0;
                                                                     }
          if(t > period)
                    t = 0.0:
         //glLineStipple(1, 0xFFEE);
//glEnable(GL_LINE_STIPPLE);
         //glBegin(GL POINTS);
         for(; ; t += step)
                    if(t > period)
                    //if(t > 3 * period)
                              break;
                              t = 0.0;
                    y2 = amp * sin(omega * t);
                    if(cache && !(cnt % 16))
                               glBegin(GL POINTS);
                              glVertex2f(cx * 6000, cy * 6);
                               glVertex2f(t * 6000, y2 * 6);
                               glEnd();
                              glBegin(GL LINE STRIP);
                              //gIVertex2f(cx * 4000, cy * 2);
                              //glVertex2f(cx * 4000, 0);
                              glVertex2f(t * 6000, y2 * 6);
                              glVertex2f(t * 6000, 0);
                              glEnd();
                    }
```