Spring 2015 **ECE 3056**

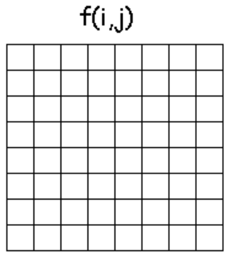
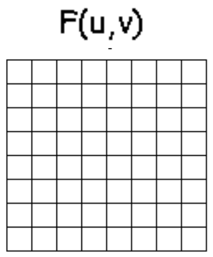
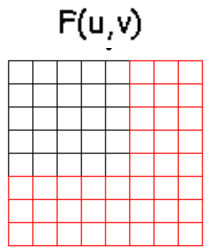
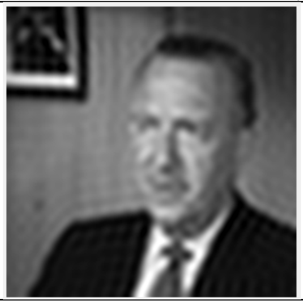
Assignment 2:Due 11:55 pm February 9th 2016

Part I (100 pts): Power vs. Image Reconstruction Error

Purpose: To gain familiarity with thinking about tradeoff's involving power consumption. Dynamic Power consumption depends on switching activity. Here we will think about dynamic power consumption in a more abstract manner (how much computation is necessary) instead of thinking about individual logic/circuit elements.

Background: Image processing is a pervasive and compute intensive task, used in applications ranging from facebook/instagram photos to youtube videos. Delivering high fidelity image processing with low power consumption is a common challenge. Raw image content typically involves large amounts of data which is typically compressed with formats like JPEG. These formats convert the original image from the spatial domain to the frequency domain using transforms such as the Discrete Cosine Transform (DCT). These transformations reveal that most of the information in the signal/image can be stored in a small number of coefficients (allowing us to throw away the rest and save space). Representing an image with a smaller amount of data means that performing operations like filtering require less computation (and therefore less power consumption). However, if too many coefficients (you will explore how many) are thrown away, the signal reconstruction deteriorates and it can become unrecognizable (pictures below).

The DCT of an N x N pixel image produces an N x N matrix. Elements of the DCT matrix represent coefficients in the frequency domain. Typically it is possible to discard (set to 0) higher frequency coefficients without losing too much information/degrading image quality. The image is recovered by taking an inverse DCT (which produces another N x N matrix representing the image in the spatial domain). This works because most images have information concentrated in lower frequency components. Beyond that, we (humans) cannot visually perceive the higher frequencies beyond a certain limit.

Original Image: N x N values representing each pixel of the image in spatial domain(i,j are pixel co-ordinates)	**DCT:** N x N matrix of values representing the image in frequency domain (u,v are frequency co-ordinates)	**Truncated DCT:** N-K x N-K matrix of coefficients (black) left un-touched. The 'K' discarded (set to 0) coefficients shown in red.
		 K ≠ 3 (red blocks-illustration only)
Images shown on right reconstructed from the respective un-truncated and truncated DCT transforms illustrated in the row above		

Spring 2015 **ECE 3056**

<u>**NOTE:** You do not have to do any image processing in this assignment. Nor do you need to have taken ECE 2026 to do this assignment. Everything you need in terms of image processing related guidance (Formulae, libraries) will be provided. You are NOT required to implement the DCT (or inverse DCT) on images</u>

<u>**Goal:**</u> Develop an energy model to compute the dynamic power dissipation and reconstruction error (%) when running a generic image filtering task with a given image. The model should allow you to configure how many terms of the DCT you wish to discard (K) and calculate corresponding power consumption and reconstruction error. You will use this model to trade off power consumption (reducing the number of coefficients) against reconstruction quality (% age error in reconstructed image) and find a good balance.

Calculating Power:
1. Each mult operation/instruction costs 10 fJ (fJ = 10^{-15} J) of energy.
2. A hypothetical filter (no actual filtering is needed) which is another N – K x N- K matrix.
3. The truncated DCT (N – K x N – K) is multiplied by the hypothetical filter
4. The image is processed in 1/30th of a second (regardless of what 'K you choose).

Calculating Reconstruction error:
1. **Formula**[1]: $Reconstruction\ Error\ (\%) = 100 * \sqrt{\frac{\sum_1^N((Image\ DCT - Truncated\ DCT)^2)}{\sum_1^N(Image\ DCT^2)}}$

With an accurately implemented model, you can study the tradeoff between power consumption and image quality with different 'K'. For example, you can use it to figure out the lowest reconstruction error achievable within a given power budget OR alternatively, figure out the minimum amount of power needed to support a maximum tolerable reconstruction error.

<u>**Requirements**</u>:
Write a C program that will
- Read a csv file specified by the user that contains the DCT of an image
- Read a csv filename specified by the user that it will write the truncated DCT to
- Read an integer 'K', the number of terms that will be truncated from the original DCT
  - You don't need to change the size of the matrix, you can just set all the truncated terms to 0 when writing the truncated DCT to the filename specified in 2.
- Calculate and print the power consumption and reconstruction error for the truncated DCT corresponding to the input parameters.

<u>**You are given**</u>
1. A <u>make</u> file
2. A Skeleton C program (<u>EnergyModel.c</u>), which takes the following inputs
   a. csv filename from which it will read the DCT of an image (files mentioned in 3.)
   b. a csv filename where it should write the truncated DCT
   c. an integer K which denotes the number of terms that should be truncated from the DCT.
3. Additional Resources [**DO NOT SUBMIT THESE – 20 point penalty if you do**]
   a. 6 csv files which contain the DCT's of the images mentioned above (for use as 2.a)
   b. [Optional Use]: 6 tiff files which contain the respective images mapping to 3

[1] If you are curious about the math and have not taken ECE 2026 - see Parseval's Theorem

c. [Optional Use]: The following MATLAB functions and script

i. ConvertImgToDCT.m: Reads an image file. Shows image. Takes DCT of image and writes it to a specified filename. Use it if you wish to experiment with images beyond those provided. Looking for more images? Check here or here

ii. ConvertDCTFileToImg.m: Reads DCT (truncated or otherwise) from a CSV file (like the kind your program will generate), applies inverse DCT and renders the image in a MATLAB figure. Use it to see how images look after truncation.

iii. ECE3056_Assignment2_SampleScript.m: Simple script which calls the above functions in sequence for a file 'Cronkite.tiff'.

Submission Guidelines

Please submit the following in a zip file named "EnergyModel.zip". **Do not submit anything else**.

- Your make file
- Your completed C program (EnergyModel.c)
- PDF Report (described below in Grading Guidelines)

Grading Guidelines

We'll be checking whether your model produces correct results. To do so, we'll be using some images (not given to you) and some truncation parameters we select during grading. We'll compare the output of your model with our model solution.

- Program compiles without errors on Klaus linux labs: **25 points**
- Program Produces results in reasonable ranges (not perfectly accurate, but close): **25 points**
- Program produces precisely correct results: **25 points**
- Report with your findings and program documentation, description and comments: **25 points**
  - Your code has your name on top and meaningful comments which help TA's follow it
  - You submit a report (PDF file) along with your code which has
    - Your name and GTID
    - A Concise (**0.5 - 0.75 page max**) and clear description of how your program works.
    - A brief (**1 page max)** summary which includes results from using the model to explore the parameter space (i.e for different images, what happens to energy/reconstruction values at different truncation levels/ K's). Your report should include at least one 2 axis plot (here is how to do them in excel OR matlab, whichever you prefer) which overlays reconstruction error AND energy consumption (y axes) with different 'K's on the x axis and a qualitative explanation of your observations from doing this analysis. This analysis MUST be done for at least one image, but you are welcome to evaluate more images and share your observations/additional plots if you wish (viewed favorably for grading).

Logistical Advice: The code provided works on Klaus Linux labs machines. This is where it will be graded (NO EXCEPTIONS) using the makefile and EnergyModel.cpp only. It is your responsibility to make sure that your code works in that environment prior to submission. You can code in the lab or on your Mac, Windows, Linux machines in any preferred environment (VS, Eclipse etc), but please test that everything works in the lab (DON'T ASSUME).

Part II (50 pts): Problem Set

(**15 pts.**) Write the MIPS assembly equivalent of following C code:

```
// Sample C program snippet
// Swapping two numbers without using a temporary variable

int main()
{
        int x = 10, y = 5;

        // code to swap 'x' and 'y'
        x = x + y; // x now becomes 15
        y = x - y; // y becomes 10
        x = x - y; // x becomes 5

        printf("After Swapping: x = %d, y = %d\n", x, y); // use QTSPIM I/O here
        return 0;
}
```

2. (**25 pts**) Answer the following questions with respect to the MIPS program shown below. Assume that the data segment starts at 0x10010000 and that the text segment starts at 0x00400000.

```
        .data
label:  .word 24, 28
        .byte 64, 32
        .asciiz "Example Program"

        .text
main:   jal push
        jal pop
        li $v0, 10
        syscall

pop:    lw $fp, 0($sp)
        lw $ra, 4($sp)
        add $sp, $sp, 32
ret1:   jr $ra

push:   subiu $sp, $sp, 32
        sw $fp, 0($sp)
        sw $ra, 4($sp)
ret2:   jr $ra
```

a) Write the values of the words stored at the following memory locations. Provide your answer in hexadecimal notation.

Word Address	Value
0x10010008	
0x0040000c	
0x00400000	

b) Consider the process of assembly of the above program,

i) What would the symbol table in the output file contain?

ii) What relocation information would be recorded if any?

c) what are the values of the following labels?

ret1:

push:

d) Assume that the procedure push and pop were assembled in a distinct file and statically linked with the main program. Further assume that the procedures (starting with the instruction labeled pop) are placed in the memory starting at location 0x00400040. Provide the encoding of the jal pop instruction.

3. (**10 pts**) Assume that all register-register instructions except jump instructions consume10nJ of energy while jump instructions consume 20nJ. All memory instructions consume 40nJ. Consider the following program

```
# a simple counting for loop and a conditional if-then-else statement

        .data
L1:     .word 0x44, 22, 33, 55   # array for which we will compute the sum

        .text
        .globl main

main:   la $t0, L1              # initialize starting address
        li $t1, 4                         # initialize loop count
        add $t2, $zero, $zero   # initialize sum

loop:   lw $t3, 0($t0)          # load first element
        add $t2, $t2, $t3       # update sum
        addi $t0, $t0, 4        # point to next word
        addi $t1, $t1, -1       # decrement count
        bne $t1, $zero, loop    # check if done

        bgt $t2, $0, then       # if the sum >0, move sum to $s0
        move $s0, $t2
        j exit
then:   move $s1, $t2           # else move sum to $s1

exit: li $v0, 10
        syscall
```

1. For the preceding program
    a. How much energy is expended in the execution of the above program.
    b. What is the average power dissipation if the processor is executing at 1 GHz