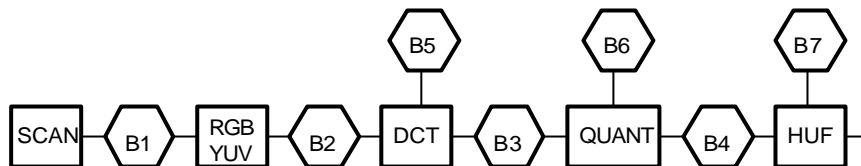


MEMORY MANAGEMENT IN THE JPEG 422 BASELINE ENCODER

The first step in the creation of a custom architecture is the design of the memory architecture. In this homework, you will think of and design a custom memory-architecture for the JPEG 422 Baseline Encoder. At this point in the design flow, we are interested in the overall system architecture. We use a system model as shown below.



1. SCAN is a block that reads out a pixel stream in from a X (horizontal) by Y (vertical) picture. The pixel stream consists of consecutive bytes of (R,G,B) data.
2. RGBYUV is a block that converts a pixel in the (R,G,B) color space to a pixel in the (Y,U,V) color space.
3. DCT performs a 2-D Discrete Cosine Transform on a block of 8-by-8 pixels.
4. QUANT quantizes a block of 8-by-8 pixels.
5. HUF reads in a stream of pixel values and performs run-length coding.

Each of the blocks B1 through B6 is a buffer structure with three aspects: a write address sequence, a buffer size, and a read address sequence. At this point we ignore the synchronization between the operation blocks like SCAN and the buffers like B1, and assume that all reads and writes in buffers will never violate a data precedence. Some of the buffer structures are read-write (like B1), while others are read-only (like B6). A read-only buffer has no write-sequence.

**THE ASSIGNMENT**

Using the information provided in the class presentation and the JPEG reference code available from the class project homepage, find for each of the buffers size and the read-write addressing mechanism. Look in the C++ reference code to identify each of the blocks and buffer structures. Each of those will be mapped into a variable or a variable member of an object. For complex buffers, more than one variable might be used.

Once you understand the flow of data in the code, complete the table as shown below. The goal is to identify, for each buffer, a write sequence, a read sequence, and the size.

Example:

Buffer	Contents	Type	Read	Write
B8	128 locations, 32 bit	Read/Write Random Access	Read by XYZ, with address sequence 0- >127	Read by OPQ, with address sequence 127->0
...	...	...	...	...

**GETTING STARTED**

Check the reference C++ code for JPEG on the project homepage. Install it on your computer and compile it. It will help you later in the project if you spend some time on this now. Check the presentation slides for JPEG again. By understanding how the JPEG algorithm works, you will be able to fill out the table easily.

**WHAT TO TURN IN**

Due Date: 4/18/2002 at 12 o'clock noon at Letty's desk (7440 Boelter Hall).

What: 1 sheet containing the following

- The table with buffer characteristics mentioned above