24.422 SWITCHING & AUTOMATA THEORY

LABORATORY 3

DESIGN OF A GRAPHICS PROCESSOR: PART 1: THE DPU

PURPOSE & OBJECTIVES

The purpose of this lab is to implement the Data Path Unit (DPU) of a graphics controller called **GRAFIX** for use in a simple handheld gaming console. This laboratory will lead to a CCU implemented in Lab 4 and a completion of the project in Lab 5.

The objective is to learn how to

(a) formulate an architecture of a simple graphics processor,
(b) formulate a DPU,
(d) formulate an appropriate ALU, and
(c) describe and verify them.

MAJOR ACTIVITIES

1. Study GRAFIX of Figs. 1 and 2.
2. Study the operation set for the minimal ALU.
3. Complete the architecture of the GRAFIX system.
4. Complete the architecture of GRAFIX unit.
5. Complete the architecture of the DPU in GRAFIX.
6. Describe and verify ALU and DPU of GRAFIX using Verilog.
7. Follow the instructions of TA in the lab.

TIME TO COMPLETE: 3 hours

ANTICIPATED BACKGROUND

1. 24.222 Logic Circuits.
2. 24.361 Microprocessing Systems.
3. 24.424 Microprocessor Interfacing.
4. 24.423 Digital Systems Organization (now 24.376 Digital Systems Design I).

RESULTS

An optimal DPU in GRAFIX.

LAB REPORT

See Section 7 of this lab writeup.

ACKNOWLEDGEMENT

Jonathan Greenberg, **Alexis Denis**, and **Luotao Sun** have helped in the design and preparation of this laboratory.

1. INTRODUCTION

The purpose of this lab is to implement the Data Path Unit (DPU) of a graphics controller for use in a simple handheld gaming console.

2. DESCRIPTION OF GRAFIX

2.1 What is GRAFIX?

GRAFIX is an 8-bit graphics controller designed to interface a host machine with a frame buffer (video memory) which is then displayed on a 240x180 pixel screen. As shown in Fig. 1, the graphics controller receives instructions (command stream) from a host processor, and outputs results to a frame buffer. This frame buffer consists of a 240 x 180 x 8 memory array. Each byte represents a gray-scale value (called here COLOUR) of a pixel on the display.

This graphics controller should perform two basic functions:
(a) plotting a pixel into the frame buffer, and
(b) plotting a line in the frame buffer using Bresenham's scan conversion technique.

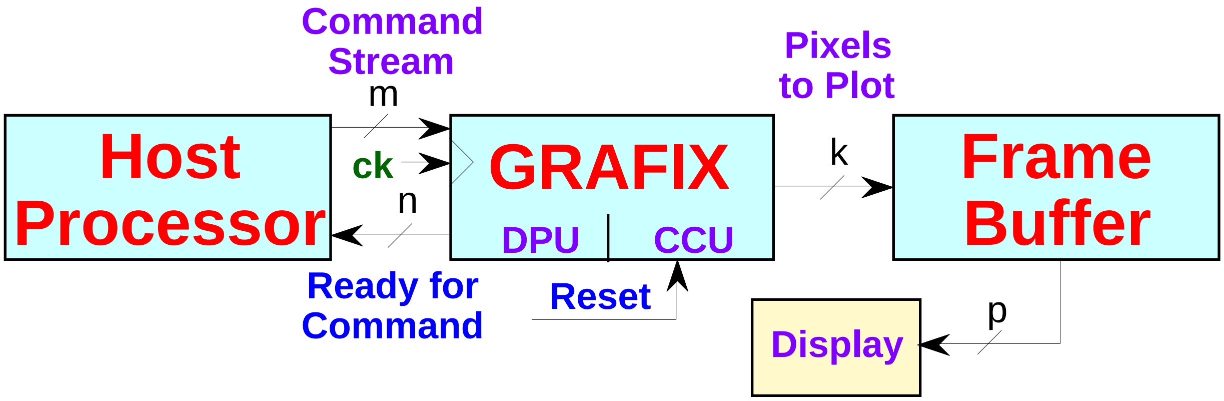

Fig. 1. A host controls a display through GRAFIX and a frame buffer.

2.2 Description of GRAFIX

As usual, GRAFIX can be partitioned into the two major parts: (a) a data path unit (DPU), and (b) a computing control unit (CCU). The DPU is responsible for data transformations, while the CCU provides the control and sequencing functions for the DPU. The system has a clock and a reset mechanism so that it enters into a known state during initialization. As shown in Fig. 2, the DPU of GRAFIX is comprised of several blocks (units), including:

(a) input/output buffers/drivers;
(b) instruction and result registers (Ri and Rr);
(c) an arithmetic-logic unit, ALU;
(d) a condition code register (CC);
(e) a register file (register memory);

(f) internal inteconnecting buses; and
(g) external inteconnecting buses.

2.3 Description of CCU

The instruction register, Ri, receives the instruction stream, passes each instruction to the command interpreter for decoding, and to the controller/sequencer which generates control sequence to execute the command.

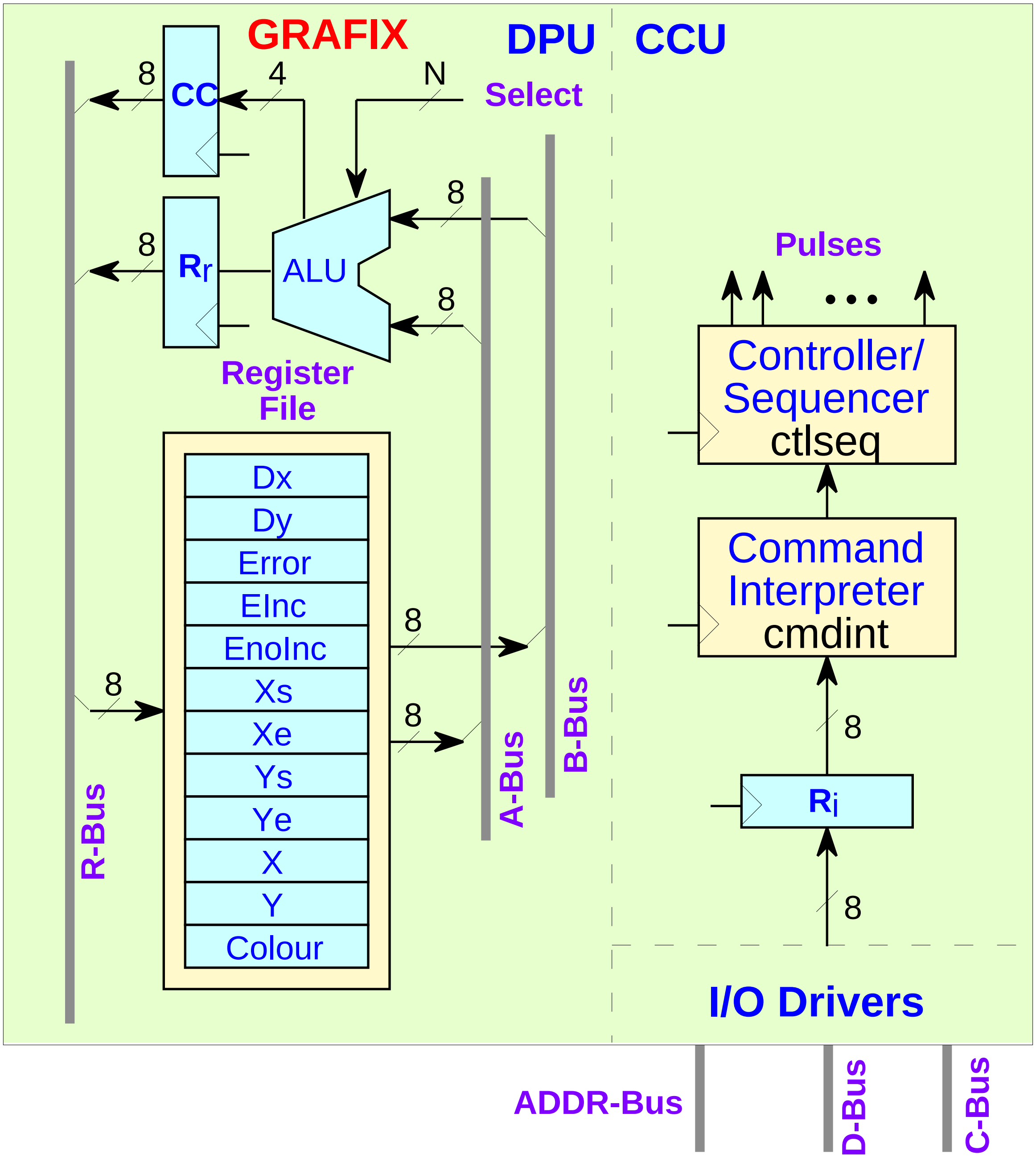

Fig. 2. Block diagram of GRAFIX.

The Instruction Register

The instruction register is necessary because we do not know if the input provider (the host) can hold the lines stable for the entire duration of the clock cycle when both the instruction interpretation and execution sequencing occur.

2.3 Description of DPU

The ALU

The main goal of this lab is to design an appropriate ALU for the DPU of GRAFIX. In a general design problem, the detailed functionality of the ALU would have to be developed from ground up. For uniformity of this lab, we shall specify the minimum number of functions that the ALU must perform, as shown in Table 1.

Table 1: ALU Operations.

No.	Operation	Operation Code
0	Signed 8-bit addition (A + B)	
1	Signed 8-bit subtraction (A – B)	
2	Multiply A by 2 (shift left by 1 position; keep the sign)	
3	Divide A by 2 (shift right by 1 position; keep the sign)	
4	Pass A to output	
5	Pass B to output	
6	Maximum of A or B (select the greater value of A or B, and pass the value to the result bus)	
7	Minimum of A or B (select the smaller value of A or B, and pass the value to the result bus)	

Notice that while an ALU such as the 74LS181 has 32 functions (16 arithmetic and 16 logic) our ALU requires only 8 operations.

The Result Register

The result register is necessary to assure that the output is also stable for the duration of the clock cycle. The two registers introduce, however, a delay (latency) of up to two clock cycles between the input and output.

The CC Register

The condition code register hold the following result flags after each operation of the ALU:

(a) Carry-Out/Borrow-Out, C;
(produced by each addition, subtraction and multiplication);
(b) Zero flag, Z,
(also produced by each addition, subtraction and multiplication, with TRUE iff the Rr is zero);
(c) “greater than”, G,
(produced by each MAX or MIN);
(d) “less than”, L,
(produced by each MAX or MIN)

The Register File

The DPU has 11 8-bit registers to store the following parameters for a line to be drawn on the display:

Xs, Xe	the starting point of the line
Ys, Ye	the ending point of the line
Dx	Dx = Xe – Xs
Dy	Dy = Ye – Ys
X,Y	a current point on the line being drawn
Colour	the colour of the line (constant for the entire line)
Error	error between two candidate points and the ideal line in the X coordinate
EInc	error for the case for which Y increases during the approximation process
EnoInc	error for the case for which Y does not increase during the approximation process

Notice that Dy/Dx is the slope of the line. The coordinates `X` and `Y` are passed on to the frame buffer, and so is the `Colour` parameter.

The Internal Buses

The DPU has a triple-bus organization, with its two input buses (A-Bus and B-Bus) and one result (output) bus (R-Bus). This is the fastest bus configuration, allowing both operands arrive at the ALU at the same time, and the separate result bus is totally isolated from the ALU, thus preventing races in the circuit.

2.4 Description of Connections to GRAFIX

The External Buses

We shall group the external connections to GRAFIX into the common triple-bus organization: (a) address bus (ADDR-Bus), (b) data bus (D-Bus), (c) control bus (C-Bus).

3. BRESENHAM'S ALGORITHM

3.1 The Procedure

The Bresenham algorithm is as follows.

```
procedure Bresenham(Xs,Ys,Xe,Ye,Color);
var
    X,Y,Dx,Dy,Error,EInc,EnoInc: Integer;
begin
    Y := Ys;
    X := Xs;
    Dy := Ye - Ys;
    Dx := Xe - Xs;
    EnoInc := Dy shl 1;
    Error := EnoInc - Dx;
    EInc := Error - Dx;
    while (X<=Xe) do
        begin
            X := X + 1;
            OutputPixel(X,Y,Colour);
            if Error < 0 then
                Error := Error + EnoInc;
            else begin
                Y := Y + 1;
                Error := Error + EInc;
            end;
        end;
    end;
end;
```

Notice that this procedure works for slopes Dy/Dx = 0 to 1 only.

4. PROBLEM 1: GRAFIX ARCHITECTURE

Investigate the block diagram of Fig. 1 and the GRAFIX unit of Fig. 2.

4.1 GRAFIX System Architecture

The system shown in Fig 1 is incompletely specified. Make the following decision regarding the machine specifications:

(a) Interfacing of the host to the GRAFIX unit may be synchronous or asynchronous.
Select one, and discuss the consequences of your choice.
(b) The frame buffer can be either a single port memory or a dual port memory.
Select one, and discuss the consequences of the choice.
(c) Interfacing of the GRAFIX unit to the frame buffer may be synchronous or asynchronous.
Select one, and discuss the consequences of the choice.

4.2 GRAFIX Unit Architecture

The GRAFIX unit shown in Fig. 2 is incomplete. Make final selections in the design, and complete the following elements:

(a) Decide on the number of operation select lines, N, to the ALU;
(b) Where do the Select lines come from?
(c) Specify the function of the I/O buffers drivers, and select appropriate devices.
(d) Provide a connection to the DPU from the external buses;
(e) Provide a connection from the internal output registers `X`, `Y`, and `Colour` to the external buses;

4.3 External Bus Architecture

The external buses to GRAFIX are not defined.

(a) Specify the number of lines on the ADDR-Bus;
(b) Specify the number of lines on the D-Bus; and
(c) Specify the number of lines on the C-Bus, and define each line.

5. PROBLEM 2: ALU IN VERILOG

Implement and verify the ALU in GRAFIX using Verilog.

6. PROBLEM 3: DPU IN VERILOG

Implement and verify the DPU OF GRAFIX using Verilog.

7. REPORT

- Describe the additions and modifications for Problem 1.
- Provide the solution and procedure for Problem 2.
- Provide the solution and procedure for Problem 3.

- Can you answer the questions in the POINTS TO PONDER section?
- Have you reached the objectives of this lab?

REFERENCES

[HaSo96] Gary D. Hachtel and Fabio Somenzi. *Logic Synthesis and Verification Algorithms.* Boston, MA: Kluwer, 1996, 564 pp. (ISBN 0-7923-9746-0)

[Kins99] W. Kinsner. *Switching & Automata Theory*. Course Notes for 24.422, Winnipeg MB: University of Manitoba, 1999.

POINTS TO PONDER

GRAFIX

1. Describe the GRAFIX system.
2. Describe the CCU in GRAFIX.
3. Describe the DPU in GRAFIX.
4. Describe the ALU in GRAFIX.
5. Describe the internal and external buses in GRAFIX.

COMPLETENESS OF GRAFIX

1. Is the GRAFIX system completely defined?
2. With your modifications, is the system synchronous or asynchronous?
3. Where does the clock come from?
4. Where does the reset signal originate?

The ALU in GRAFIX

1. Are the ALU's eight operations sufficient for GRAFIX?
2. Can we reduce the number of elementary operations in the ALU? At what cost?

The DPU in GRAFIX

1. Is the result register, Rr, absolutely necessary for proper operation of the system?
2. What are the options for passing the contents of the X, Y, and Colour registers to the frame buffer?
3. What are the options to communicate between the host and GRAFIX?
4. Is the DPU optimal?