UNIVERSITI KUALA LUMPUR
MALAYSIAN INSTITUTE OF INDUSTRIAL TECHNOLOGY

FINAL EXAMINATION
JANUARY 2016 SEMESTER

COURSE CODE	:	JCB 30304
COURSE TITLE	:	DATA ACQUISITION SYSTEM
PROGRAMME LEVEL	:	BACHELOR
DATE	:	31 MAY 2016
TIME	:	9.00 AM – 12.00 PM
DURATION	:	3 HOURS

INSTRUCTIONS TO CANDIDATES

1. Please read the instructions given in the question paper CAREFULLY.
2. This question paper is printed on both sides of the paper.
3. This question paper consists of ONE (1) sections.
4. Answer FOUR (4) ONLY questions in Section A.
5. Please write your answers on the answer booklet provided.
6. LabVIEW Palette are enclosed as reference.
7. Please answer all questions in English only.

THERE ARE 15 PAGES OF QUESTIONS EXCLUDING THIS PAGE.

SECTION A (Total: 100 marks)

INSTRUCTION: Answer 4 (FOUR) questions ONLY.
Please use the objective answer sheet provided.

Question 1

(a) LabVIEW is a data-flow programming language. A data-flow language is significantly different than text based programming languages like C, C++, or Java. Compare text based programming and graphical programming on the *execution*, *syntax* and *error checking*.

(3 marks)

(b) Figure 2 shows the front panel of a function that verify whether a given number is odd or even. Create the block diagram.

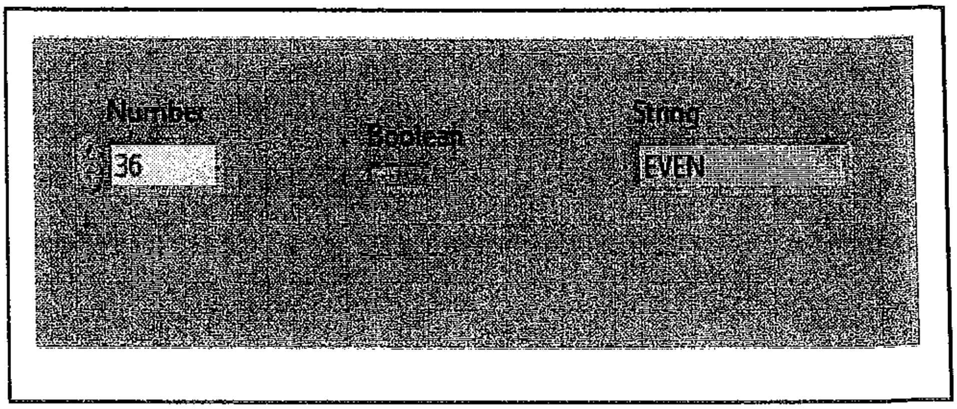


Figure 1 Front Panel for Question 1(b)

(5 marks)

(c) Figure 1 is the block diagram of an example VI. Identify the LabVIEW objects in Table 1 by giving the name of the object, the data type; checking the Object Type, i.e. function(F), control (C), indicator (I), Constant (Co).

Complete Table 1 based on your analysis from Figure 1.

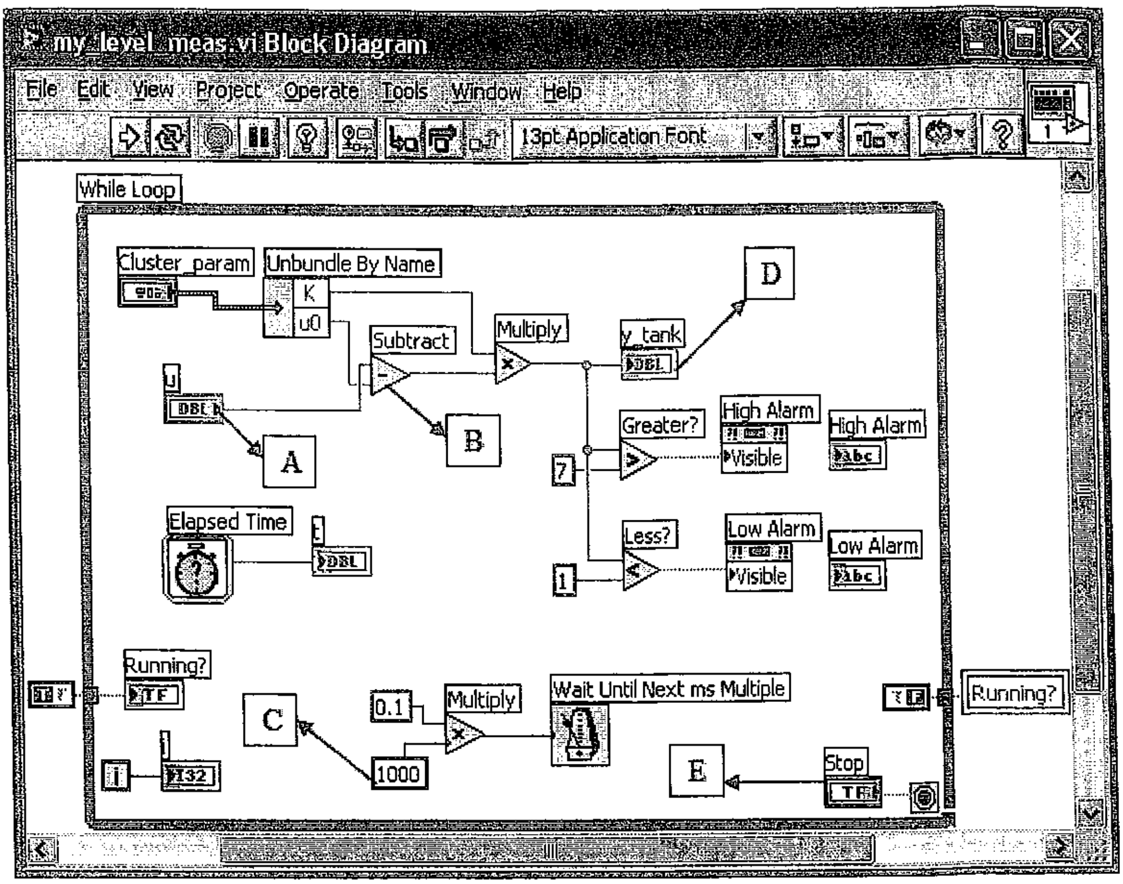


Figure 2 Block Diagram for Question 1(c)

Table 1 LabVIEW Objects

OBJECT	OBJECT NAME	OUTLINE COLOUR	DATA TYPE	OBJECT TYPE			
				F	C	I	Co
A		Orange					
B		Orange	-				
C		Blue					
D		Orange					
E		Green					

(7 marks)

(d) Although LabVIEW is a development environment built around a graphical programming language, LabVIEW also allows you to create .m files and work with text-based math.

(i) List two methods user can use Mathscript in LabVIEW.

(ii) Create a Mathscript to plot a function of graph as per below equation:

$$x = -1 + 0.02 * t$$

$$y = \sin(0.2 * t)$$

The output for the script is as per Figure 3.

(Hint: Create script from Matchscript Window)

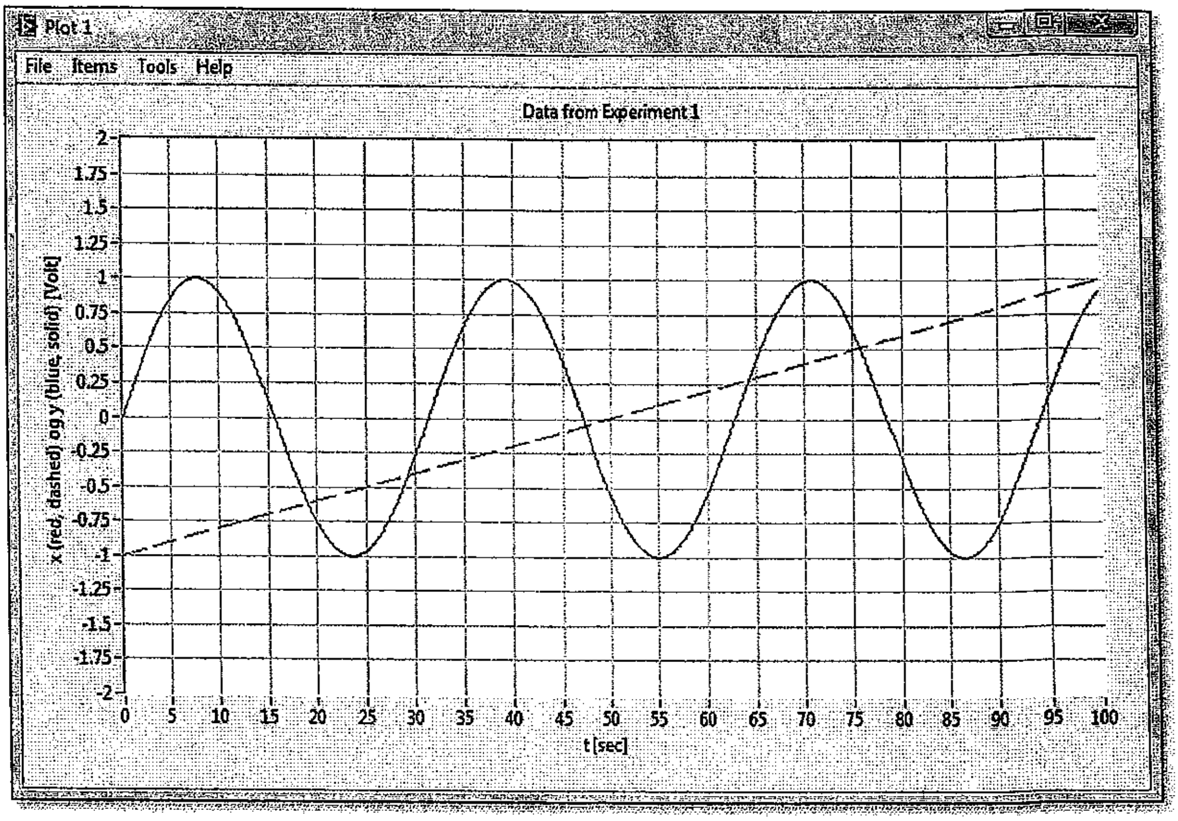


Figure 3 Output for Question 1(d)

(10 marks)

Question 2

(a) Boolean objects simulate switches, push buttons, and LEDs. Perform Boolean Operations of NAND and XOR. The front panel as per Figure 4.

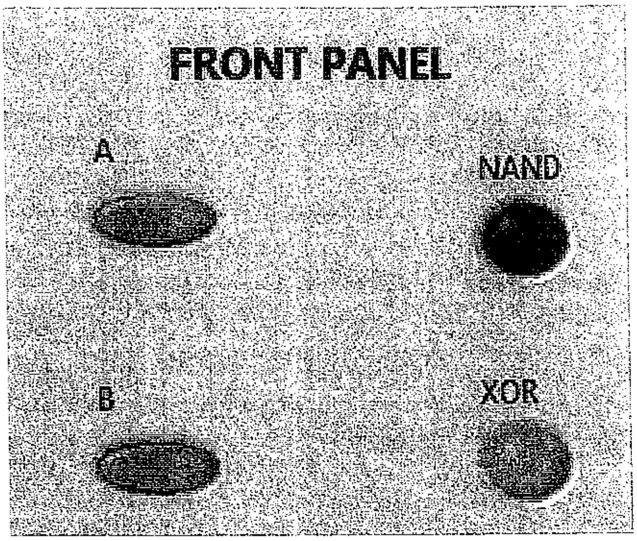


Figure 4 Front Panel for Question 2(a)

(1 marks)

(b) LabVIEW provides several tools to help get your VIs running, and running correctly. In your words, define below terms:

(i) Broken wires

(ii) Broken Run Button

(iii) Run button

(iv) Execution Highlighting

(4 marks)

(c) The While loop is an iteration construct that executes until a predetermined Boolean value is passed to its conditional terminal. Referring to the While loop structure, explain:

(i) Function of While Loop

(ii) Iteration terminal

(iii) Conditional terminal

(iv) The two settings for the Conditional Terminal

(5 marks)

(d) Figure 5 shows the frequency resonant circuit. In simple reactive circuits with little or no resistance, the effects of radically altered impedance will manifest at the resonance frequency predicted by the equation given below.

$$f = \frac{1}{2\pi}\sqrt{\frac{1}{LC} - \left(\frac{R}{L}\right)^2}$$

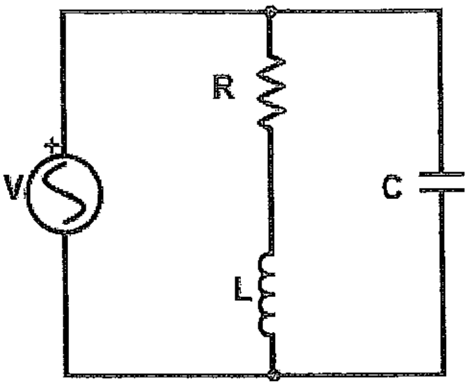


Figure 5 Frequency Resonant Circuit

Create a LabVIEW Formula Node (Front Panel and Block Diagram) to find the frequency resonance when $L = 0.15H$, $R = 14\Omega$, and $C = 2.6\,x\,10^{-6}F$. Show your answer on the Front Panel.

(8 marks)

(e) From your answer in (b), modify the design into conventional method by using fully numeric and comparison palette only. Use case structure for True and False case. Refer appendix 1, to design the block diagram.

(7 marks)

Question 3

(a) The For Loop and the While Loop can index and accumulate arrays at their boundaries automatically, adding one new element for each loop iteration. This capability is called autoindexing. By using appropriate block diagram, show the condition where the autoindexing is enable and disable.

(4 marks)

(b) A LabVIEW array is a collection of data elements that are all the same type similar to traditional programming languages. Refer to Figure 4, evaluate the output for each of the output block.

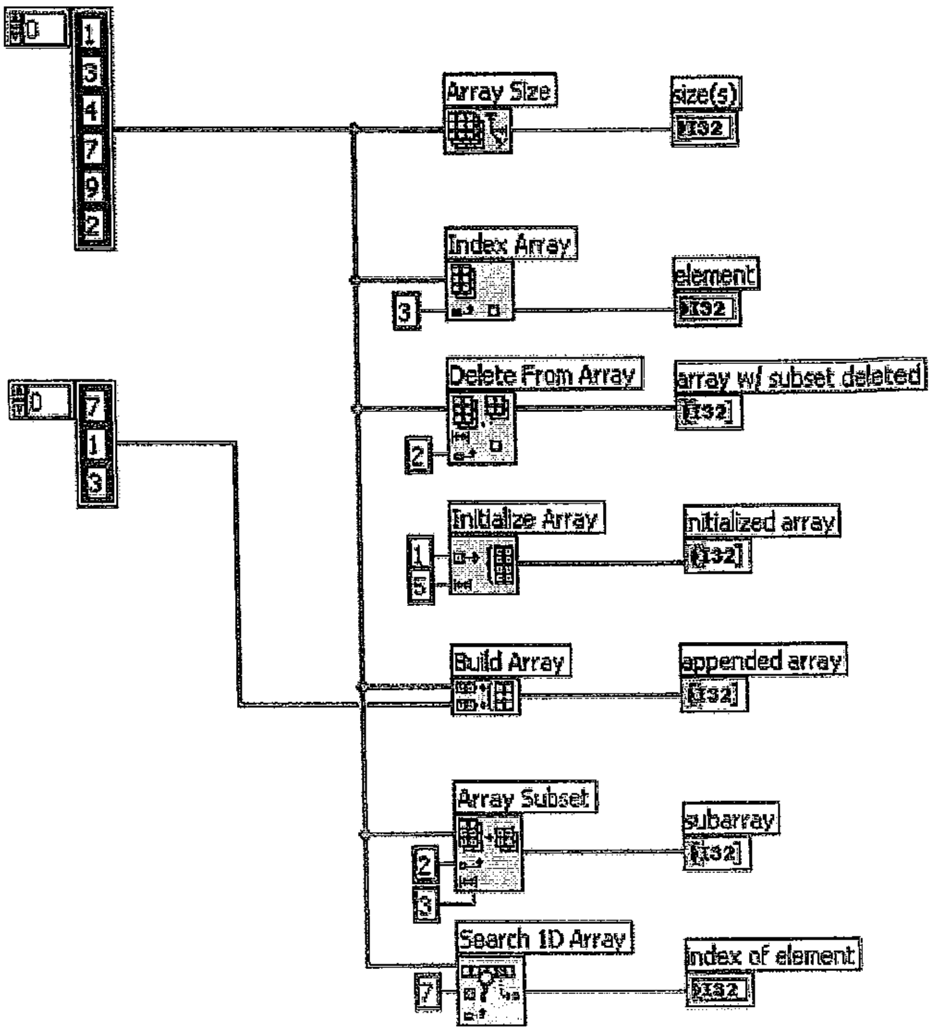


Figure 6 Components in LabVIEW for Question 3(b)

(5 marks)

(c) Ahmad want to classify whether the number is positive or not. If the number is positive then he will to double the value and display the value. If the number is negative, an error message will be display (use dialog box).

Create the VI (Front Panel and Block Diagram) to accomplish this task.

(Hint: Use Case structure)

(5 marks)

(d) Larkin Stadium have a seat of seats 15000. For final Piala Malaysia, Courtside seats sell for RM10, endzone for RM8, and balcony for RM6. The total revenue from a sell-out is RM114000. If half the courtside and balcony seats and all of the endzone seats are sold, the total revenue will be RM65000. Determine the number of seats are there in the Larkin Stadium.

Hint: Solve the problem by using simultaneous linear equations using matrix manipulation (inverse matrix).

Show step by step taken in order to rearrange the equation into matrix form. By using LabVIEW, create the Block Diagram and Front Panel.

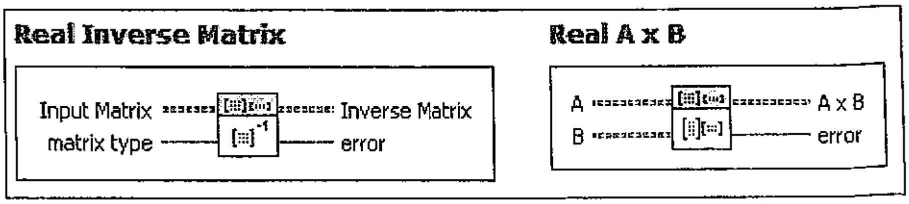


Figure 7 Numeric Palette Operator for Question 3(d)

(11 marks)

Question 4

(a) A VI can be use on the block diagram of another VI. At one hand, there is no limit on the number of layers in the hierarchy. On the other hand, using modular programming helps you manage changes and debug the block diagram quickly.

(i) In your own words, **define subVI** in LabVIEW.

(ii) Refer to Figure 8, redraw back the block diagram to simplify the coding by using subVI.

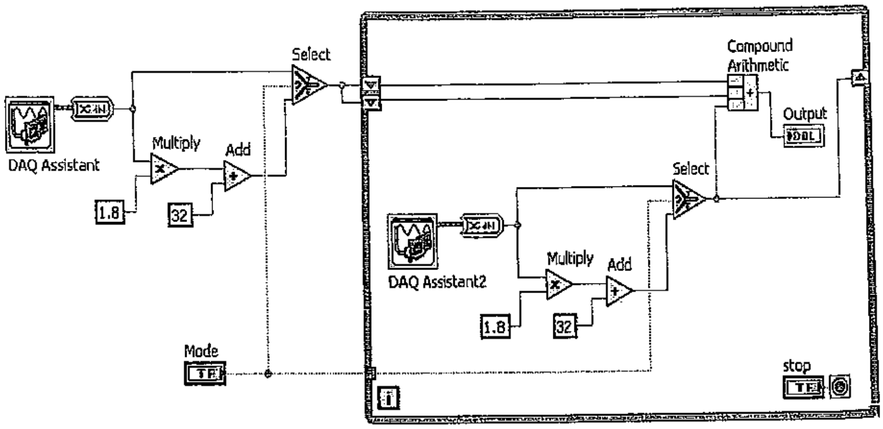


Figure 8 Block Diagram for Question 4(a)

(3 marks)

(b) Define Cluster and give example of cluster VI in LabVIEW.

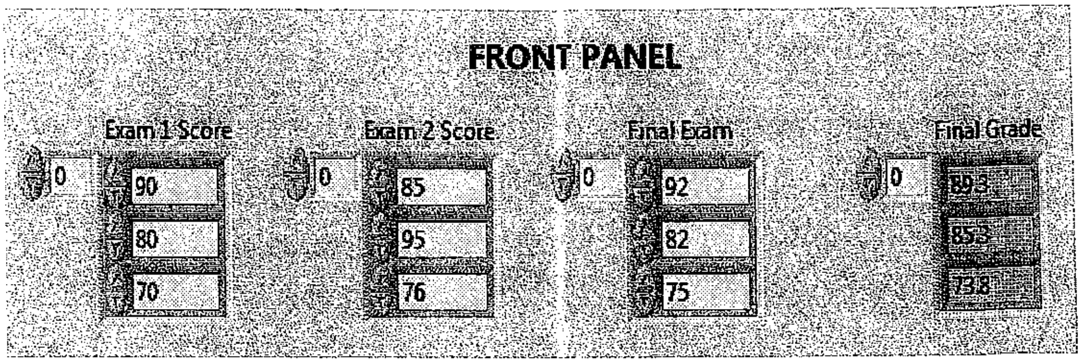
(4 marks)

(c) Explain Flat sequence structure, in term of function, structure and Data Flow. Then build a block diagram to execute the following expression:

$$Y = (A+B)^2 / (B*A)$$

(8 marks)

(d) Write a VI that compute the final grades in a course using three input arrays containing students' scores on three different exams, using the data as shown in the following front panel. The input consists of three 1D arrays: Exam 1 Scores (weighted 30% of the final grade); Exam 2 Scores (weighted 30% of the final grade); and Final Exam Scores (weighted 40% of the final grade). The VI should output a 1D array containing the final course grade for each student.



FRONT PANEL

Exam 1 Score: 0 | 90, 80, 70

Exam 2 Score: 0 | 85, 95, 76

Final Exam: 0 | 92, 82, 75

Final Grade: 0 | 89.3, 85.3, 73.8

Figure 9 Front Panel for Question 3(d)

(10 marks)

Question 5

(a) In LabVIEW, the flow of data determines the execution order of block diagram elements. Variables are block diagram elements that allow to access or store data in another location. Differentiate **TWO (2)** type of variables in LabVIEW.

(4 marks)

(b) Explain below terms:

(i) Finding errors (ii) Probe

(iii) Executing Highlighting (iv) Breakpoint

(8 marks)

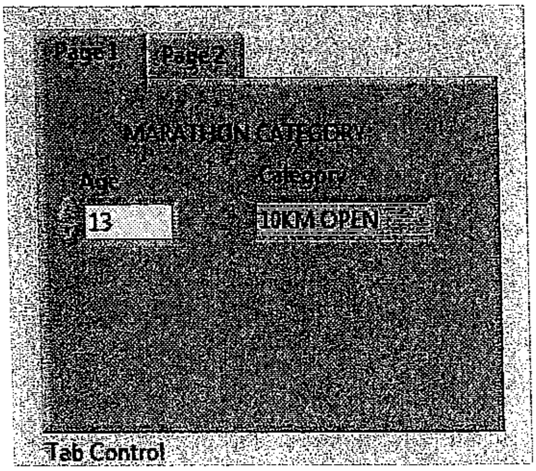
(c) Intensity graph and chart is use to display 3D data on a 2D plot by placing blocks of colour on a Cartesian plain. For example, intensity graph or chart can be used to display patterned data. Draw and explain the concept of intensity chart operation.

(6 marks)

(d) Tab control can be use to overlap front panel controls and indicators in a smaller area. Refer to Figure 10, draw the block diagram for page 1 only. The condition of the Page 1 as per below:

Category 10KM Open: Age 17 and above

Category 10KM Junior: Age between 13 to 17



Age: 13

Category: 10KM OPEN

Figure 10 Tab Control example for Question 5(d)

(7 marks)

END OF QUESTION PAPER

APPENDIX 1 FUNCTION PALETTE

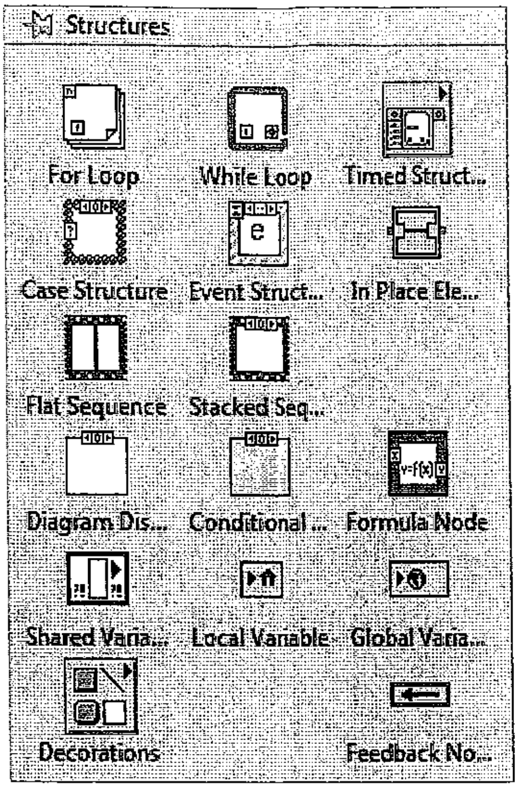


Figure 11 Structure Palette

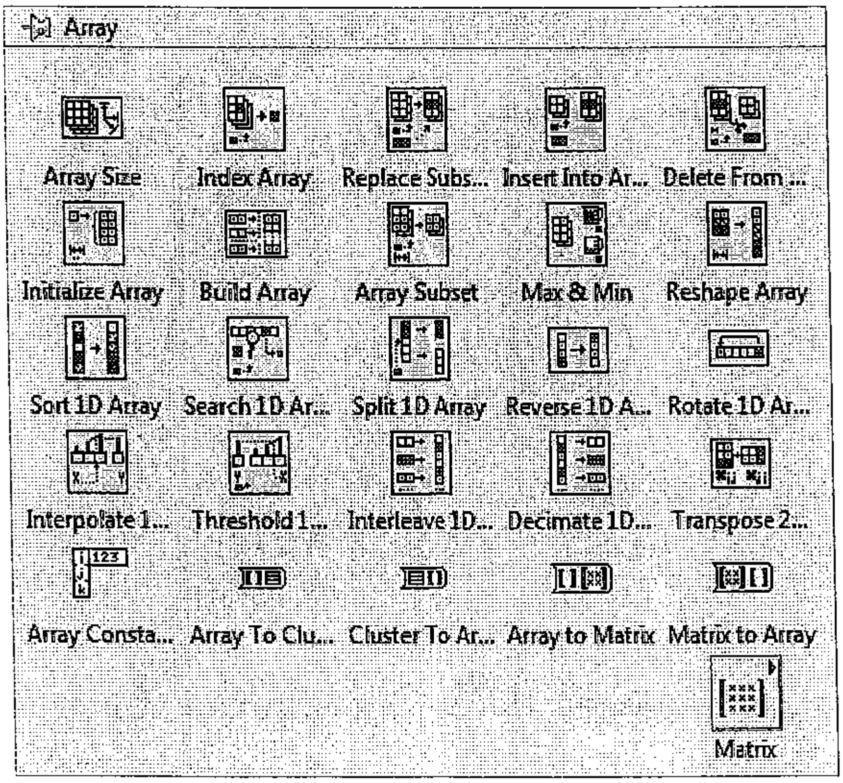


Figure 12 Array Palette

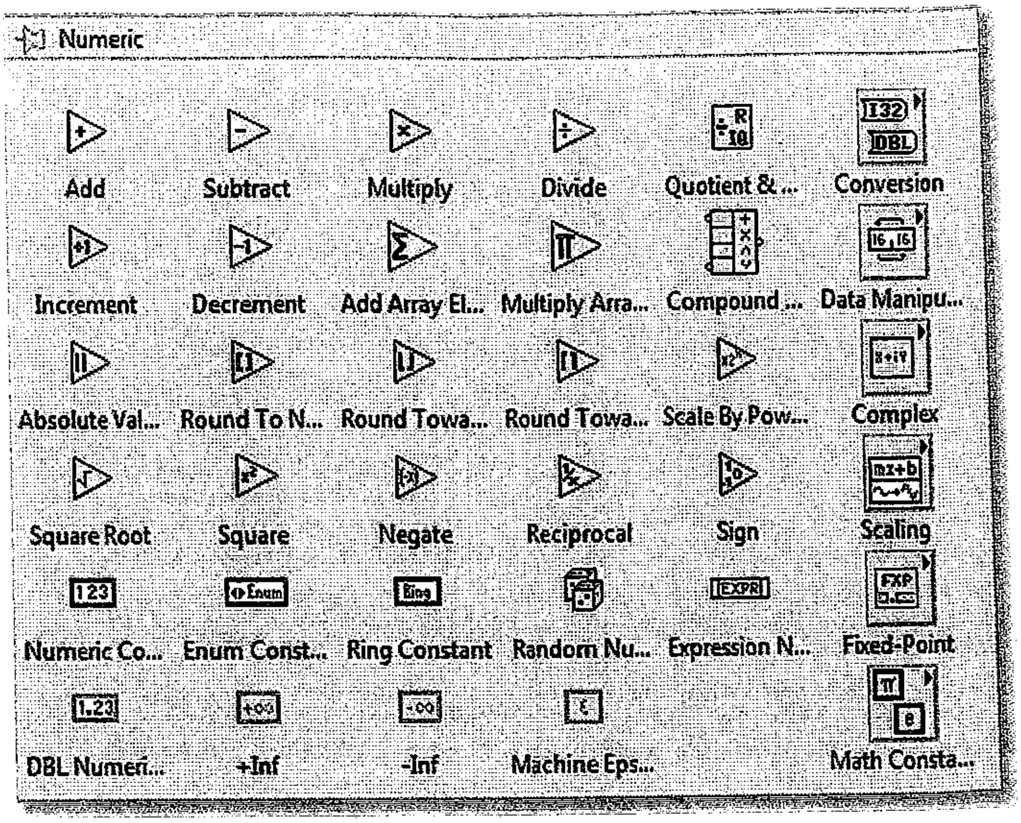


Figure 13 Numeric Palette

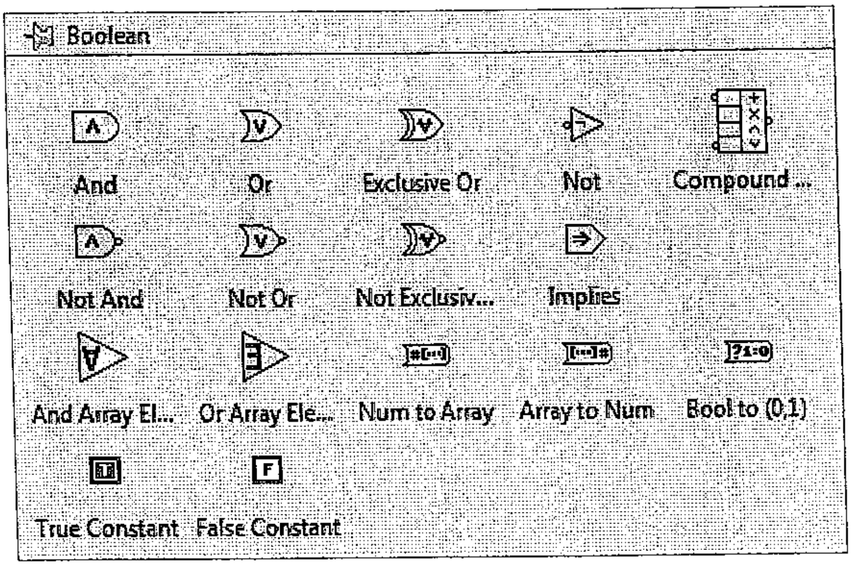


Figure 14 Boolean Palette

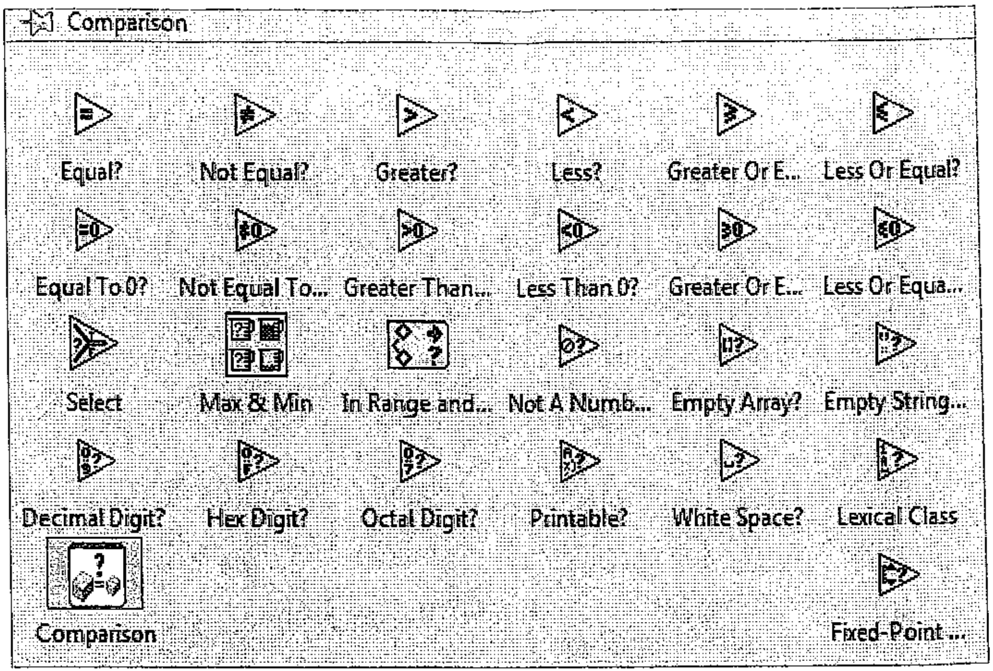


Figure 15 Comparison Palette

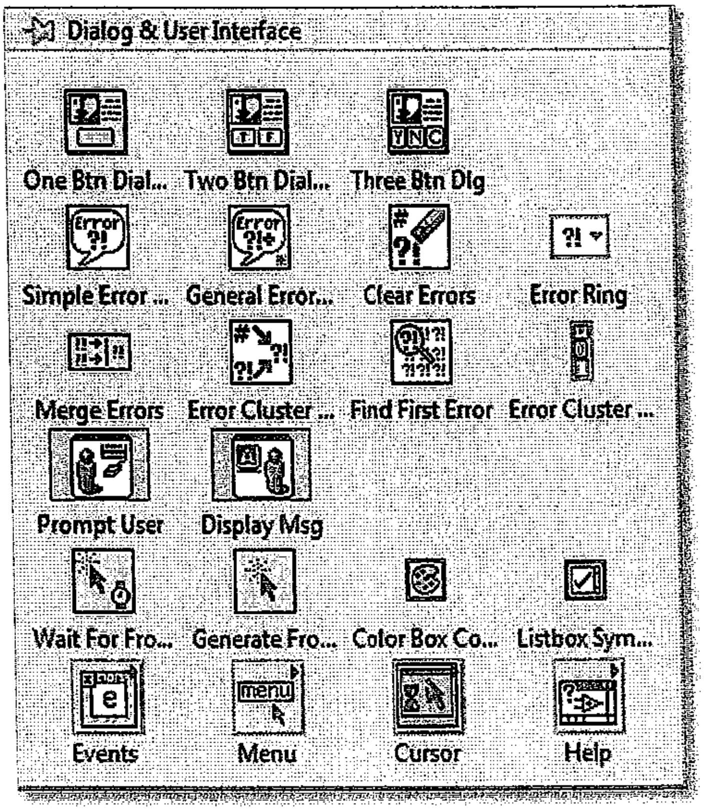


Figure 16 Dialog & User Interface Palette

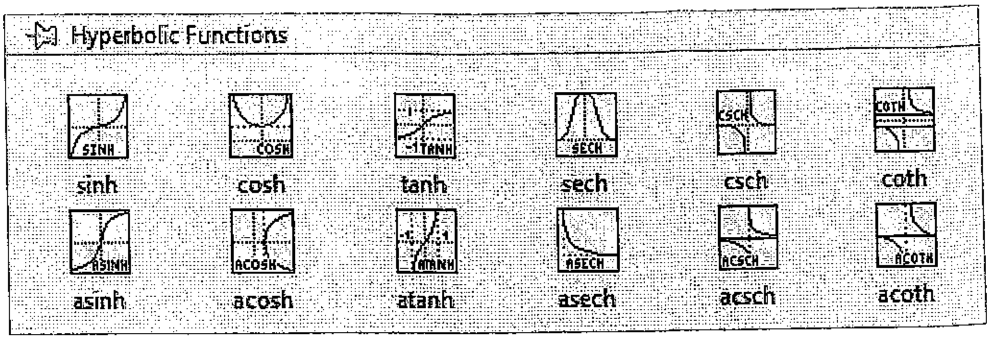


Figure 17 Hyperbolic Functions Palette

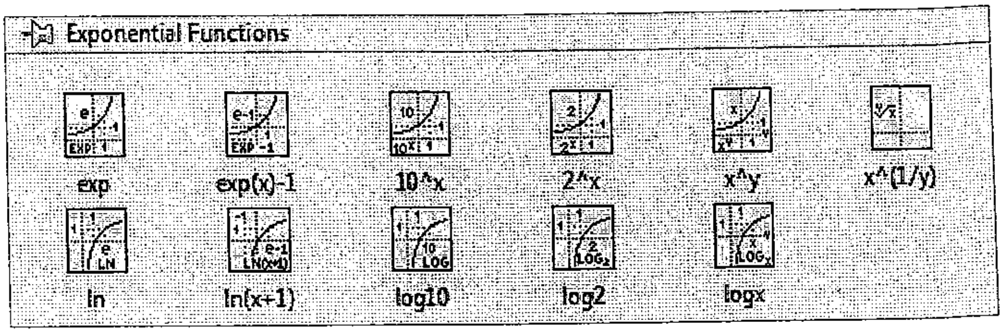


Figure 18 Exponential Function Palette

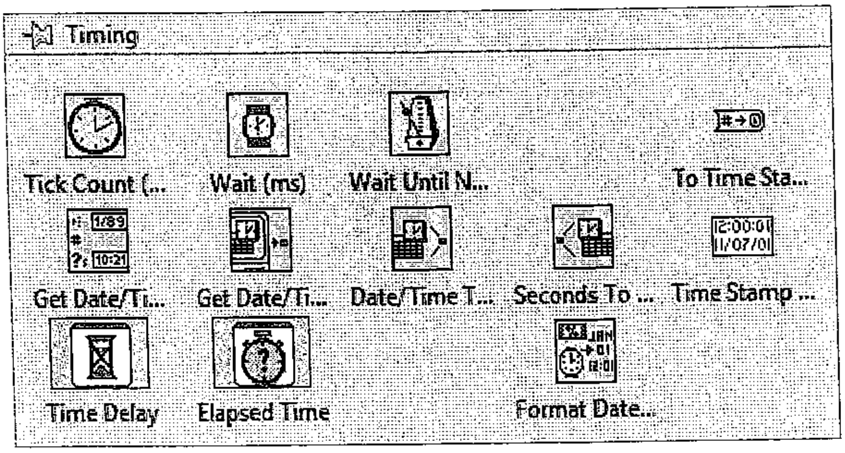


Figure 19 Timing Palette

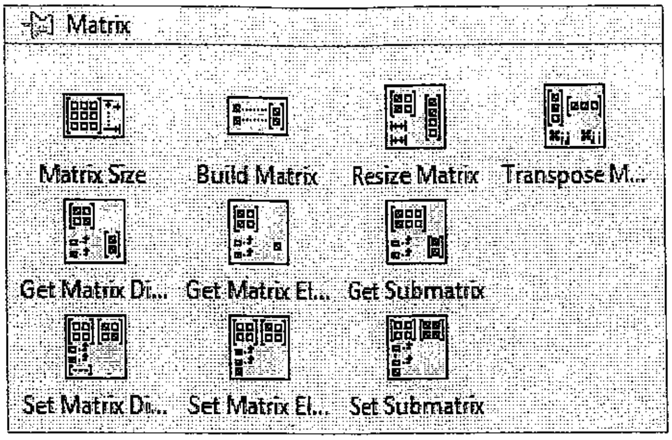


Figure 20 Matrix SubPalette

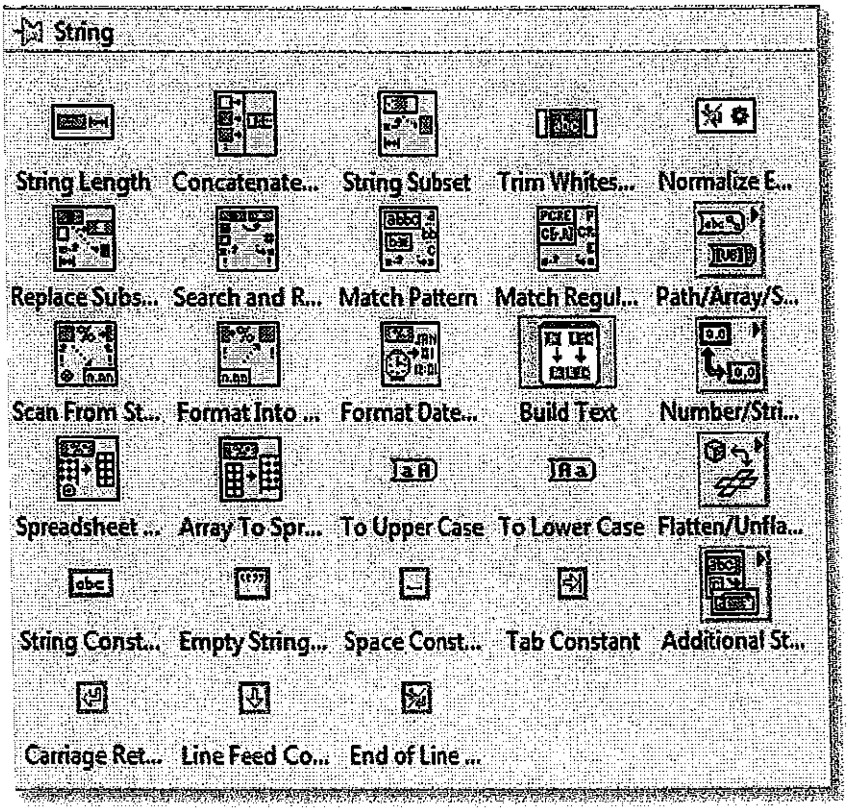


Figure 21 String Palette