

How to Prepare Block Diagrams, Flow Charts and Pseudocode

Preparation of Block Diagrams

Block diagrams describe the building blocks of complex systems. Each block is shown as a rectangle (or ellipse) with corresponding input(s) and output(s) connected by lines/arrows to other blocks of the system. The distinct blocks of the system can be represented as interconnected “black boxes” (see Fig. 1). Each box must have a short and concise label description on white background surrounded by solid black borders. Provide sufficient box size and distance among boxes within the allowed margins.

Self-explanatory images can sometimes be included in the block diagrams for more aesthetic view of apparatus and projects and to highlight the existence, availability or importance of a particular device (see Fig. 2).

Preparation of Flow Charts

Flow charts describe algorithms and programs by means of a set of graphical symbols (Alternate Process, Process, Subprocess, Data, Decision, etc.) interconnected by arrows (see Fig. 3). The flow charts follow some strict rules for their visual appearance:

- A flow chart starts with a Start symbol and ends with a Stop symbol.
- The variables and constants must be initialized before use.
- Processes/Subprocesses have only one input and one output. Decisions have only one input and two outputs, correspondingly.
- The flows of symbol sequences are oriented vertically on the page. The main flow is directed from the upper side to the lower side of the column. Secondary backward flows are directed in the opposite direction. Horizontal expansions of flow charts are due to the secondary forward and backward flows.
- A flow chart should be placed on a single column and should not continue to another column.

The flow charts must be self-explanatory and easy to interpret and analyze.

Both block diagrams and flow charts must appear as distinct digital images in the figures of the paper. The images can be prepared by using any suitable graphical software. The following formatting requirements apply to figures containing block diagrams and flow charts:

- *Size*: Figures should fit within the column width. However, large figures can span both columns.
- *Background*: Draw the figures on white background with no surrounding borders.
- *Title*: Place the title on top of the figure.
- *Figure captions*: Place the captions with 6 pt line spacing below the figure.
- *Font*: Use font Arial size 10 for the text in the figure and the captions.
- *Text wrapping*: Align the graphical content of the figures as centered images with “Inline with Text” wrapping.

Do not:

- include figure captions as part of the figures.
- put figure captions in “text boxes” linked to the figures.
- use colors unless specifically needed and will remain distinct when printed in grayscale. Color images appear only in the online version of the journal.

show both a flow chart and a pseudocode for a given algorithm. Various flow charts can be presented as pseudocode and vice versa. Flow charts can be chosen to visualize the algorithm, while the pseudocode, due to its compact form, is more beneficial in case of space limitations.

* By Dobri Atanassov Batovski, Deputy Editor, AU J.T.

(Continued on inside back cover)

(Continued from inside front cover)

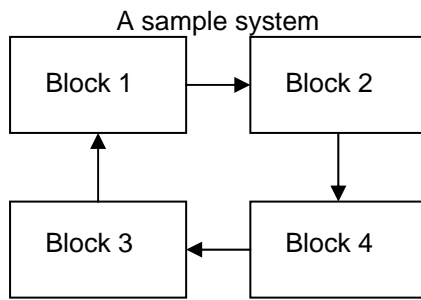


Fig. 1. A sample block diagram.

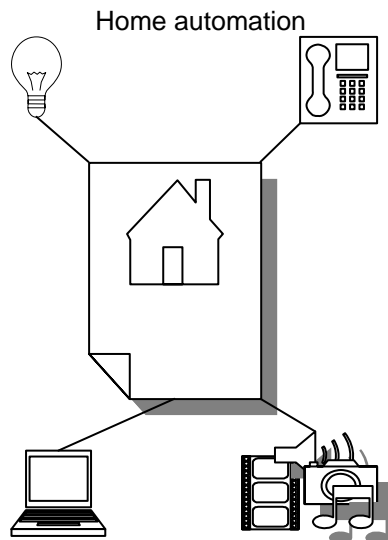


Fig. 2. A sample project.

A sample algorithm

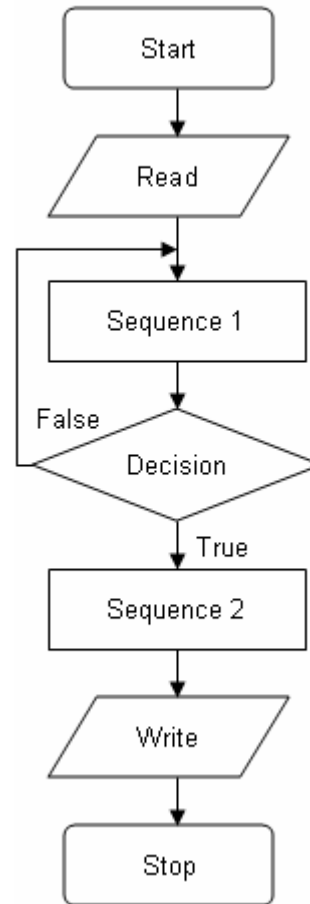


Fig. 3. A sample flow chart.

Preparation of Pseudocode

The pseudocode provides a language-independent approach of problem solving using English-like descriptions of logical constructions. The visual appearance of the pseudocode should be in a self-explanatory Input→Process→Output format. Represent in bold the reserved terms, such as **begin**, **end**, **if**, **then**, **else**, **while**, etc. The pseudocode can be shown as a part of the main text (see the sample algorithm below) or in a figure containing text (font Arial or Courier New size 10, single line spacing) on a white background surrounded by solid black borders (see Fig. 4):

```

Algorithm TITLE
/* COMMENT */
Input: INPUT PARAMETERS
begin PROCESS
1: while (LOOP CONDITION)
2:   if (CONDITION) then DO THIS
3:   else DO THAT
4:   end if
5: end while
...
end PROCESS
Output: OUTPUT PARAMETERS
  
```

```

Procedure TITLE
/* COMMENT */
Input: INPUT PARAMETERS
begin PROCESS
1: for (LOOP CONDITION)
2:   if (CONDITION) then DO THIS
3:   else DO THAT
4:   end if
5: end for
...
end PROCESS
Output: OUTPUT PARAMETERS
  
```

Fig. 4. A sample pseudocode.