



SNS COLLEGE OF ENGINEERING
(Autonomous)
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING



UNIT III EMBEDDED PROGRAMING
MODELS OF PROGRAMS



SNS COLLEGE OF ENGINEERING (Autonomous) DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING



Models of programs

Data Flow Graphs:

A **data flow graph** is a model of a program with no conditionals. In a high-level programming language, a code segment with no conditionals—more precisely, with only one entry and exit point is known as a basic block. Figure 2.14 shows a simple basic block. As the C code is executed, we would enter this basic block at the beginning and execute all the statements.

w = a+b;

x = a-c;

y = x+d;

x = a+c;

z = y+e;

Data Flow Graphs:

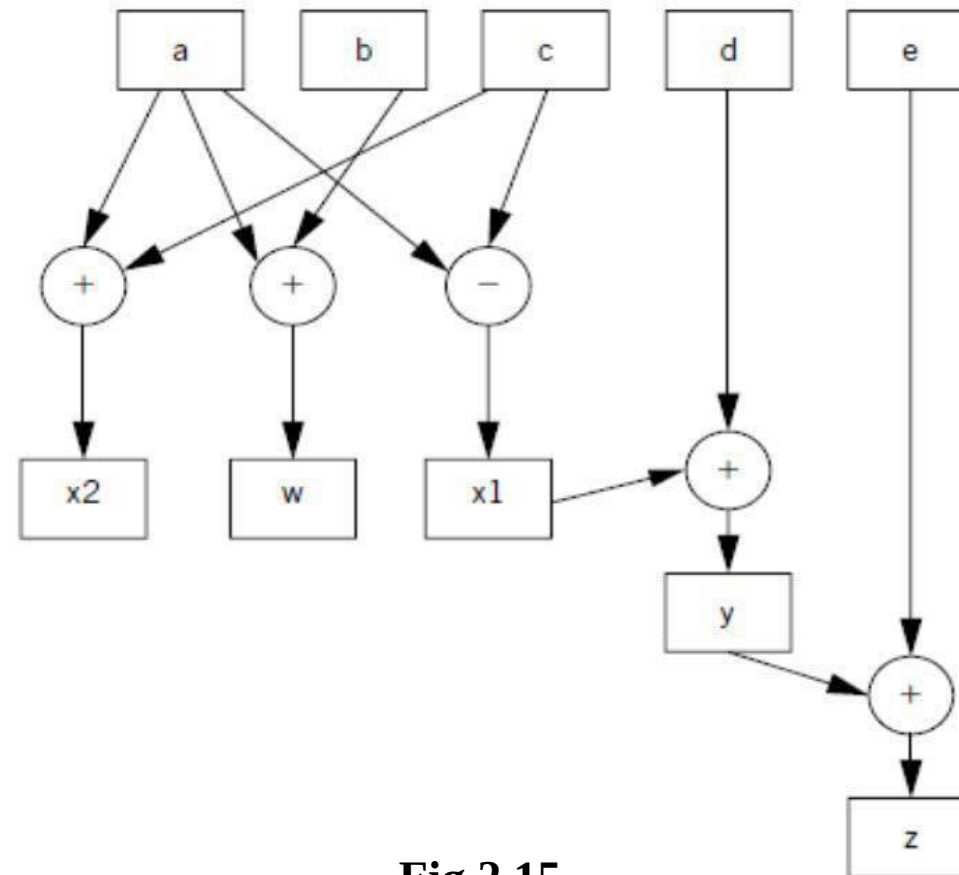


Fig 2.15



SNS COLLEGE OF ENGINEERING (Autonomous)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

basic block in C.

w = a+b;

x = a-c;

y = x1+d;

x2= a+c;

z = y+e;

The basic block in single-assignment form

- Before we are able to draw the data flow graph for this code we need to modify it slightly. There are two assignments to the variable x—it appears twice on the left side of an assignment. We need to rewrite the code in **single-assignment form**, in which a variable appears only once on the left side.
- Since our specification is C code, we assume that the statements are executed sequentially, so that any use of a variable refers to its latest assigned value. In this case, x is not reused in this block (presumably it is used elsewhere), so we just have to eliminate the multiple assignment to x. The result is shown in Figure 2.14, where we have used the names x1 and x2 to distinguish the separate uses of x.



SNS COLLEGE OF ENGINEERING

(Autonomous)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING



- The single-assignment form is important because it allows us to identify a unique location in the code where each named location is computed. As an introduction to the data flow graph, we use two types of nodes in the graph round nodes denote operators and square nodes represent values.
- The value nodes may be either inputs to the basic block, such as a and b, or variables assigned to within the block, such as w and x1.
- The data flow graph for our single-assignment code is shown in Figure 2.15. The single-assignment form means that the data flow graph is acyclic—if we assigned to x multiple times, then the second assignment would form a cycle in the graph including x and the operators used to compute x.



SNS COLLEGE OF ENGINEERING (Autonomous)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING



Control/Data Flow Graphs

A CDFG uses a data flow graph as an element, adding constructs to describe control.

In a basic CDFG, we have two types of nodes: **decision nodes** and **data flow nodes**. A data flow node encapsulates a complete data flow graph to represent a basic block. We can use one type of decision node to describe all the types of control in a sequential program. (The jump/branch is, after all, the way we implement all those high-level control constructs.

Figure 5.6 shows a bit of C code with control constructs and the CDFG constructed from it. The rectangular nodes in the graph represent the basic blocks. The basic blocks in the C code have been represented by function calls for simplicity. The diamond-shaped nodes represent the conditionals. The node's condition is given by the label, and the edges are labeled with the possible outcomes of evaluating the condition.

Building a CDFG for a while loop is straightforward. The while loop consists of both a test and a loop body, each of which we know how to represent in a CDFG. We can represent for loops by remembering that, in C, a for loop is defined in terms of a while loop.



SNS COLLEGE OF ENGINEERING (Autonomous)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING



The following for loop

```
for (i = 0; i < N; i++) {
```

```
    loop_body();
```

```
}
```

is equivalent to

```
i = 0;
```

```
while (i < N) {
```

```
    loop_body();
```

```
    i++;
```

```
}
```

```
if (cond1)
```

```
    basic_block_1();
```

```
else;
```

```
}
```



SNS COLLEGE OF ENGINEERING **(Autonomous)**

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING



```
basic_block_2();  
basic_block_3();  
switch (test1) {  
case c1: basic_block_4(); break;  
case c2: basic_block_5(); break;  
case c3: basic_block_6(): break;  
}
```

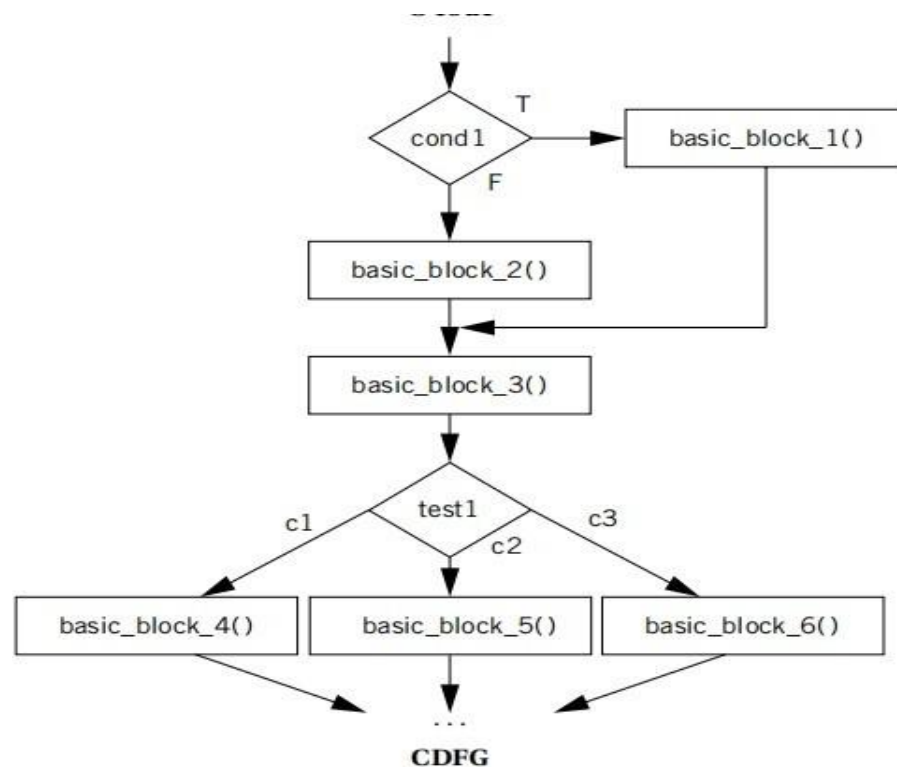


Fig 5.6 An extended data flow graph for our sample basic block



SNS COLLEGE OF ENGINEERING
(Autonomous)
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING



THANKYOU