Chapter 5

Program Design and Analysis
Figure 5.1 A circular buffer.
Figure 5.2
Figure 5.3 A producer/consumer system.
Figure 5.4 A basic block in C.

\[ w = a + b; \]
\[ x = a - c; \]
\[ y = x + d; \]
\[ x = a + c; \]
\[ z = y + e; \]
Figure 5.5 The basic block in single-assignment form.

\[ w = a + b; \]
\[ x1 = a - c; \]
\[ y = x1 + d; \]
\[ x2 = a + c; \]
\[ z = y + e; \]
Figure 5.6 An extended data flow graph for our sample basic block.
Figure 5.7 Standard data flow graph for our sample basic block.
Figure 5.8 C code and its CDFG.
while $(a < b)$ {
    $a = \text{proc1}(a, b);$  
    $b = \text{proc2}(a, b);$ 
}

C code

![CDFG diagram]

**Figure 5.9** A while loop and its CDFG.
Figure 5.10 Program generation from compilation through loading.
Figure 5.11 Symbol table processing during assembly.
Figure 5.12 External references and entry points.
Figure 5.13 The compilation process.
Figure 5.14 Flow of control in C and control flow diagrams.
Figure 5.15 Layout of a one-dimensional array in memory.
Figure 5.16 Memory layout for two-dimensional arrays.
Figure 5.17 Loop tiling.

```
for (i = 0; i < N; i++)
  for (j = 0; j < N; j++)
    c[i] = a[i,j] * b[i];
for (i = 0; i < N; i += 2)
  for (j = 0; j < N; j += 2)
    for (ii = i; ii < min(i + 2, N); ii++)
      for (jj = j; jj < min(j + 2, N); jj++)
        c[ii] = a[ii,jj] * b[ii];
```
Figure 5.18 Using graph coloring to solve the problem of Example 5.6.
Figure 5.19 A reservation table for instruction scheduling.

<table>
<thead>
<tr>
<th>Time</th>
<th>Resource A</th>
<th>Resource B</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>t + 1</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>t + 2</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>t + 3</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Figure 5.20 Code generation by template matching.
Figure 5.21 Execution time is a global property of a program.
Figure 5.22 Code motion in a loop.
Figure 5.23 Measuring energy consumption for a piece of code.
Figure 5.24 Energy and execution time vs. instruction/data cache size for a benchmark program [Li98].
Figure 5.25 The matrix representation of a graph and its basis set.
Figure 5.26 Cyclomatic complexity.
Figure 5.27 Domain testing for a pair of values.
Figure 5.28 Definitions and uses of variables.

```c
a = mypointer;
if (c > 5) {
    while (a->field1 != val1)
        a = a->next;
}
if (a->field2 == val2)
    someproc(a, b);
```
<table>
<thead>
<tr>
<th></th>
<th>Block</th>
<th>Decision</th>
<th>P-use</th>
<th>C-use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TeX</td>
<td>85%</td>
<td>72%</td>
<td>53%</td>
<td>48%</td>
</tr>
<tr>
<td>awk</td>
<td>70%</td>
<td>59%</td>
<td>48%</td>
<td>55%</td>
</tr>
</tbody>
</table>

**Figure 5.29** Code coverage of functional tests for TeX and awk (after Horgan and Mathur [Hor96]).
Figure 5.30 Frequency-shift keying.
Figure 5.31 The FSK detection scheme.
Figure 5.32 Receiving bits in the modem.
Figure 5.33 Class diagram for the modem.
Figure 5.34 Waveform generation by table lookup.

float sine_wave[N_SAMP] =
{ 0.0, 0.5, 0.866, 1,
  0.866, 0.5, 0.0, -0.5,
  0.866, -1.0, -0.866, -0.5,
  0};

**Table**
Figure 5.35 A color filter array arranged in a Bayer pattern.
Figure 5.36 The typical JPEG compression process.
Figure 5.37 Zig-zag pattern for reading coefficients.
<table>
<thead>
<tr>
<th>Name</th>
<th>Digital still camera</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Digital still camera with JPEG compression</td>
</tr>
<tr>
<td>Inputs</td>
<td>Image sensor, shutter button</td>
</tr>
<tr>
<td>Outputs</td>
<td>Display, flash memory</td>
</tr>
<tr>
<td>Functions</td>
<td>Determine exposure and focus, capture image, perform Bayer pattern interpolation, JPEG compression, store in flash file system</td>
</tr>
<tr>
<td>Performance</td>
<td>Take one picture in 2 sec.</td>
</tr>
<tr>
<td>Manufacturing cost</td>
<td>Approximately $75</td>
</tr>
<tr>
<td>Power</td>
<td>Two AA batteries</td>
</tr>
<tr>
<td>Physical size and weight</td>
<td>Approx 4 in × 4 in × 1 in, less than 4 ounces.</td>
</tr>
</tbody>
</table>

**Figure 5.38** Requirements for the digital still camera.
Figure 5.39 Structure of an EXIF file.
Figure 5.40 State diagram for display operation.
Figure 5.41 State diagram for picture taking.
Figure 5.42 Basic classes in the digital still camera.
Figure 5.43 Computing platform for a digital still camera.
Figure 5.44 Sequence diagram for taking a picture with a digital still camera.
UN Figure 5.1

Inputs/outputs
(− = no action)

Idle

No seat/−
Seat/timer on

Buzzer

No seat/buzzer off

Seated

No belt and no timer/−

Belted

No belt/timer on

Belt/buzzer off

Copyright © 2013 Elsevier Inc. All rights reserved.
UN Figure 5.2
UN Figure 5.3
UN Figure 5.4

PLC = ?? → ORG 100

label1 ADR r4,c
LDR r0,[r4]

label2 ADR r4,d
LDR r1,[r4]

label3 SUB r0,r0,r1

code

Symbol table
PLC = 100 → ORG 100

label1 ADR r4,c
LDR r0,[r4]

label2 ADR r4,d
LDR r1,[r4]

label3 SUB r0,r0,r1
PLC = 100 → label1

ORG 100
ADR r4,c
LDR r0,[r4]

label2
ADR r4,d
LDR r1,[r4]

label3
SUB r0,r0,r1

UN Figure 5.6
UN Figure 5.7

PLC = 100 → label1

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol table</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORG 100</td>
<td>label1 100</td>
</tr>
<tr>
<td>ADR r4,c</td>
<td></td>
</tr>
<tr>
<td>LDR r0,[r4]</td>
<td></td>
</tr>
<tr>
<td>label2</td>
<td></td>
</tr>
<tr>
<td>ADR r4,d</td>
<td></td>
</tr>
<tr>
<td>LDR r1,[r4]</td>
<td></td>
</tr>
<tr>
<td>label3</td>
<td></td>
</tr>
<tr>
<td>SUB r0,r0,r1</td>
<td></td>
</tr>
</tbody>
</table>
UN Figure 5.8

PLC = 104

<table>
<thead>
<tr>
<th>Label</th>
<th>Code</th>
<th>Symbol table</th>
</tr>
</thead>
<tbody>
<tr>
<td>label1</td>
<td>ORG 100</td>
<td>label1 100</td>
</tr>
<tr>
<td></td>
<td>ADR r4,c</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDR r0,[r4]</td>
<td></td>
</tr>
<tr>
<td>label2</td>
<td>ADR r4,d</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDR r1,[r4]</td>
<td></td>
</tr>
<tr>
<td>label3</td>
<td>SUB r0,r0,r1</td>
<td></td>
</tr>
</tbody>
</table>
ORG 100

label1    ADR r4,c
LDR r0,[r4]

label2    ADR r4,d
LDR r1,[r4]

PLC = 116 → label3    SUB r0,r0,r1

Symbol table

label1    100
label2    108
label3    116

Code
UN Figure 5.10
UN Figure 5.11
UN Figure 5.12

Diagram:

- **a + b > 0**
  - **T** leading to **x = 5**
  - **F** leading to **x = 7**
UN Figure 5.13

1. $a + b > 0$
2. $x = 5$
3. $x = 7$
4. 

Flowchart:
- From 1 to 3: If $a + b > 0$, then $x = 7$.
- From 2 to 4: If $x = 5$, then return to 4.

Loop initiation code

Loop test

Loop body

Loop variable update
UN Figure 5.16
UN Figure 5.17
UN Figure 5.18

```
Loop initiation code

Loop test

Loop body

Loop variable update
```
UN Figure 5.19
UN Figure 5.21a
UN Figure 5.21b