Using a simple display

Using a simple display

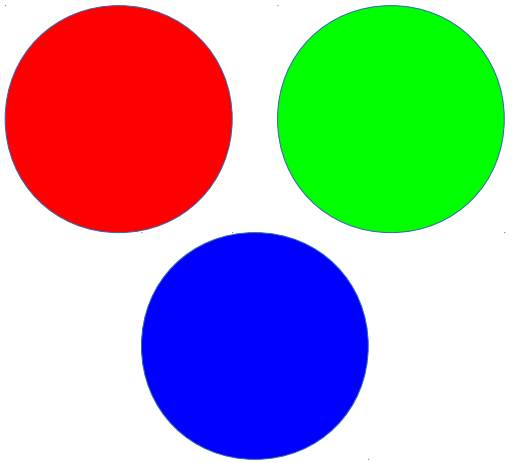
- Dot-matrix graphic display systems
  - Make use of “picture elements” - Pixels
  - Pixels can be monochrome
  - More usually colour

Using a simple display

- Bitmap display technologies
  - Cathode Ray Tube
    - Electron gun lights coloured phosphors
  - LCD
    - Polarized light is allowed pass through (to varying extents) coloured filters
  - LED
    - Each Pixel consists of a number of coloured LED's whose brightness is varied
  - OLED
    - Similar to LED, fabrication involves a print-like process. Flexible.

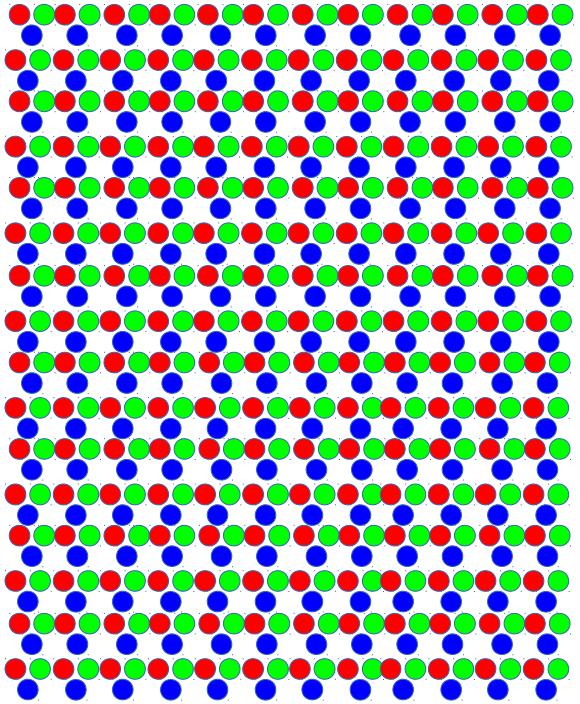
Using a simple display

- How is colour made?
  - Intensity of 3 colours is varied
  - Usually 8 bit resolution per colour
  - 2^{24} different colours possible (16 million approx)



1 Pixel

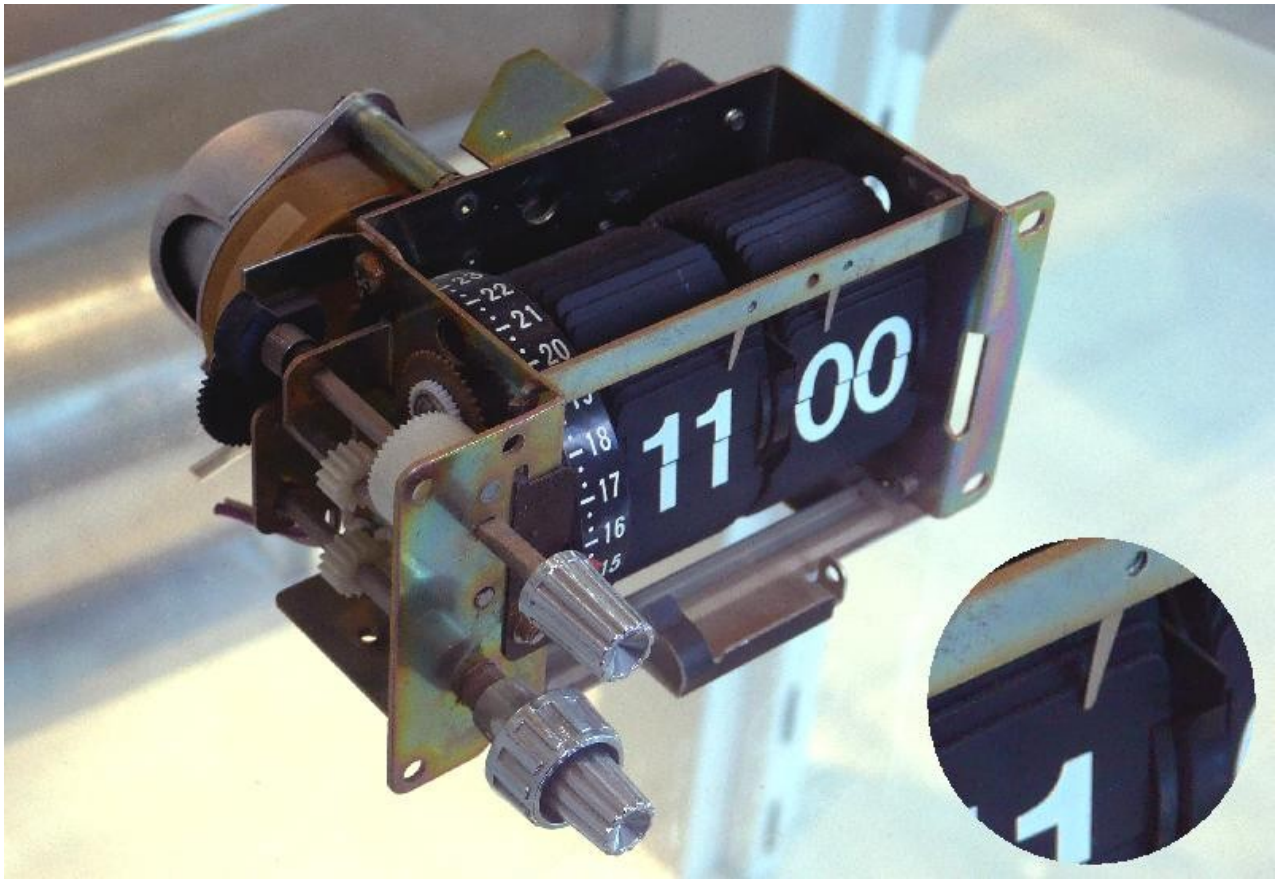
Using a simple display



- Bitmap display made up of lots of clusters of variable colour dots.
- Sometimes extra (red) dot is added per pixel for brightness
- How many pixels in a full HD TV?
- How many pixels in a 4k TV?

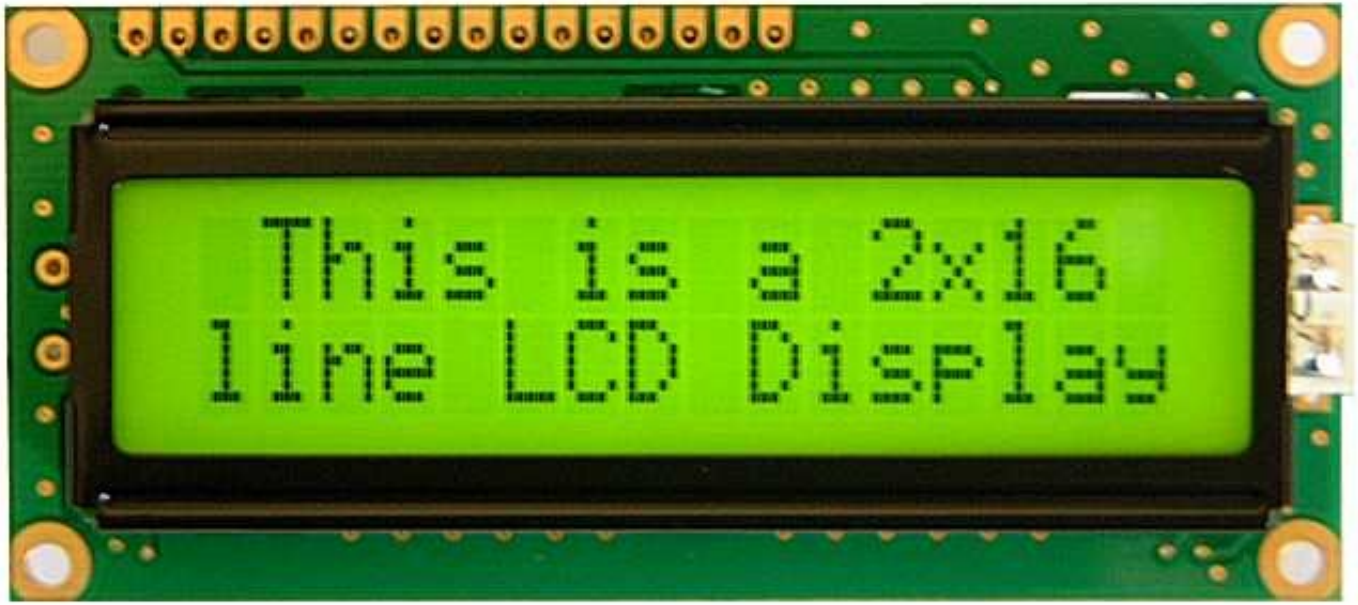
Using a simple display

- Non-Bitmap displays
  - Split flap display



Using a simple display

- Non-Bitmap displays
  - LCD display (5x8 font, intelligent)



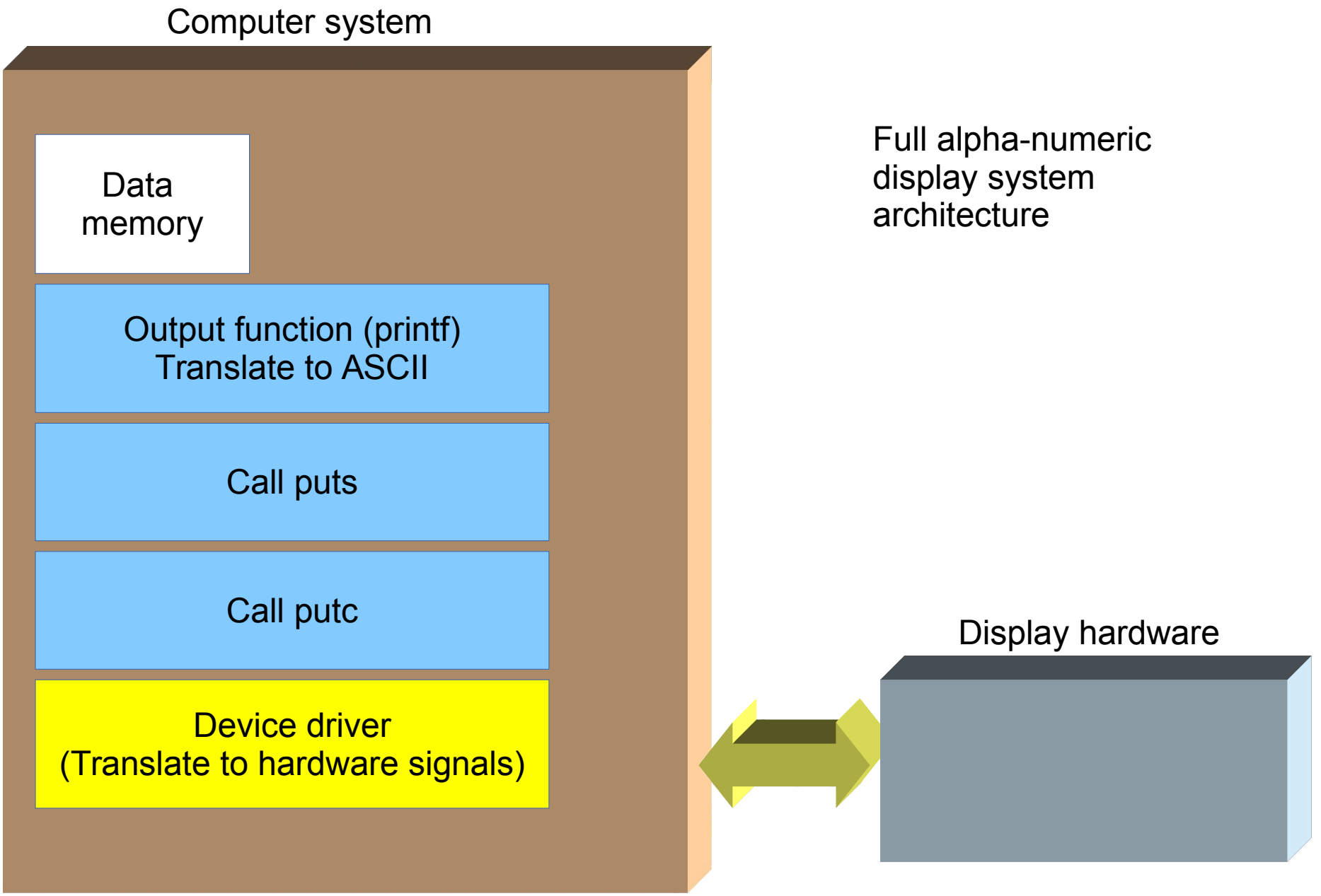
Using a simple display

- Non-Bitmap displays
  - LED display (7 segment)
  - Each segment is an LED

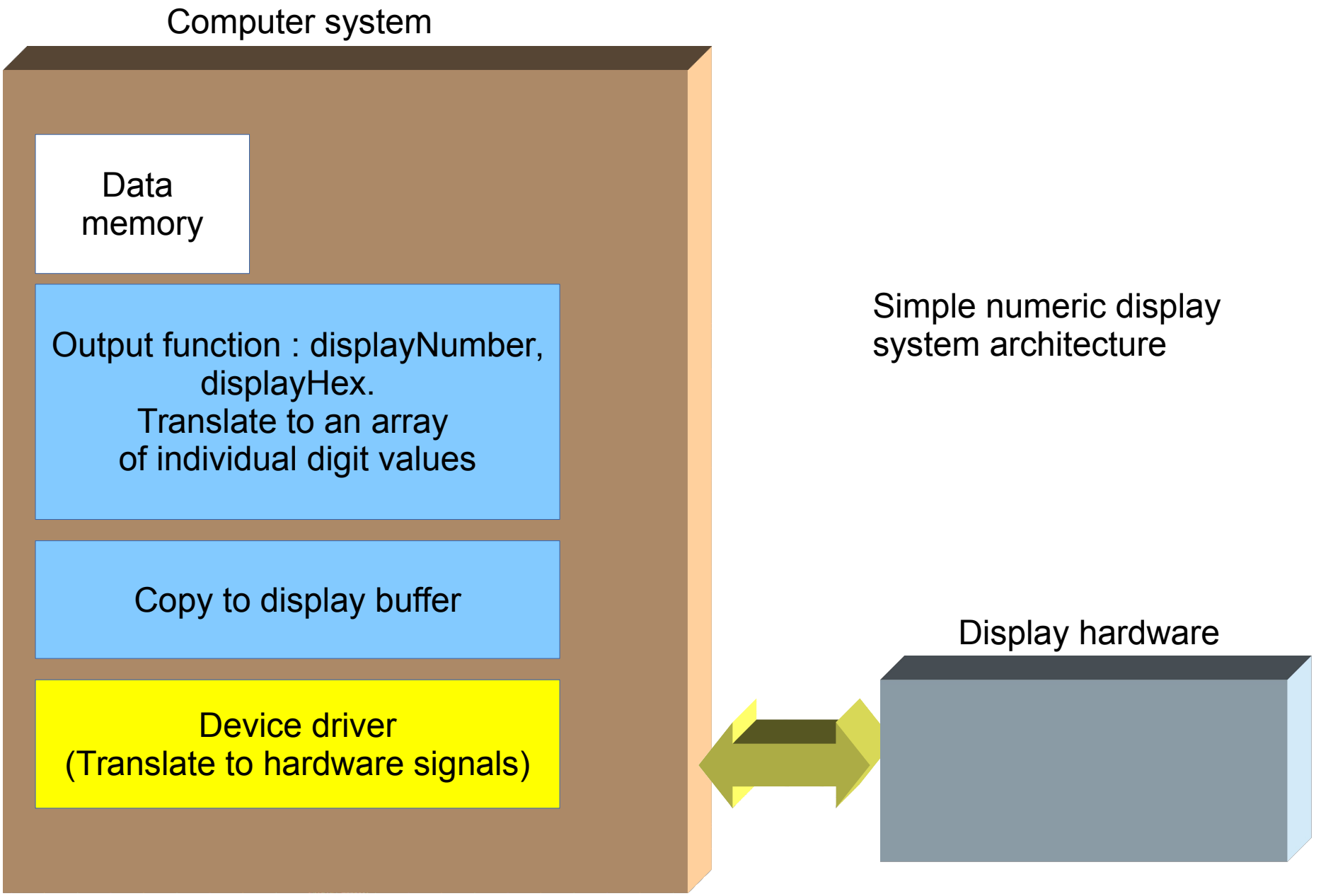
Using a simple display

- How do you get data on to the display?
  - Display represents data in a particular way
  - Computer represents data in perhaps a different way
  - Humans may have another view entirely

Using a simple display



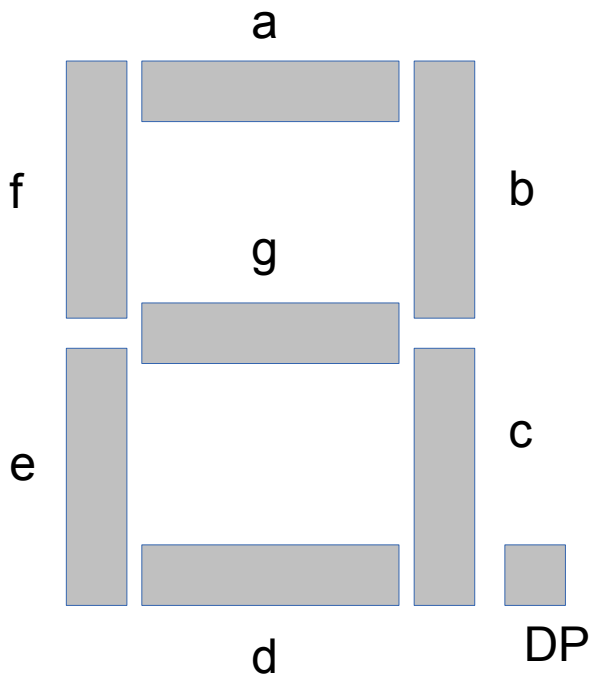
Using a simple display



Simple numeric display system architecture

Using a simple display

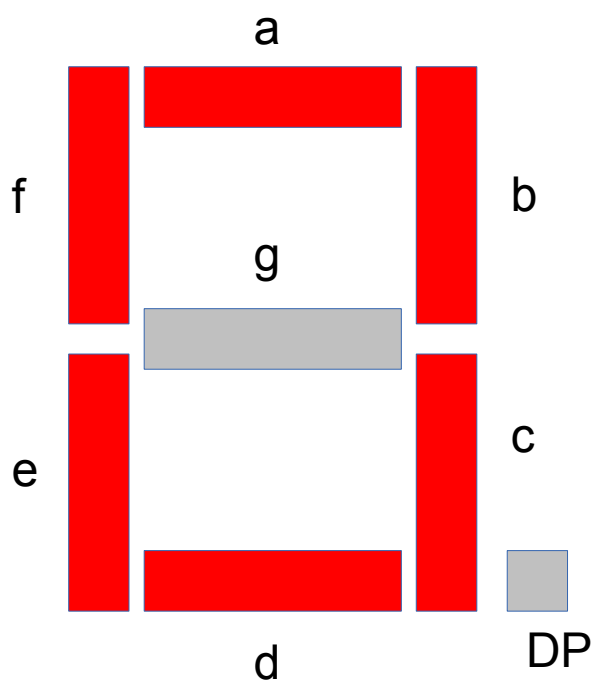
- 7 Segment display hardware



Light individual segments to make “number shapes”

Using a simple display

- 7 Segment display hardware

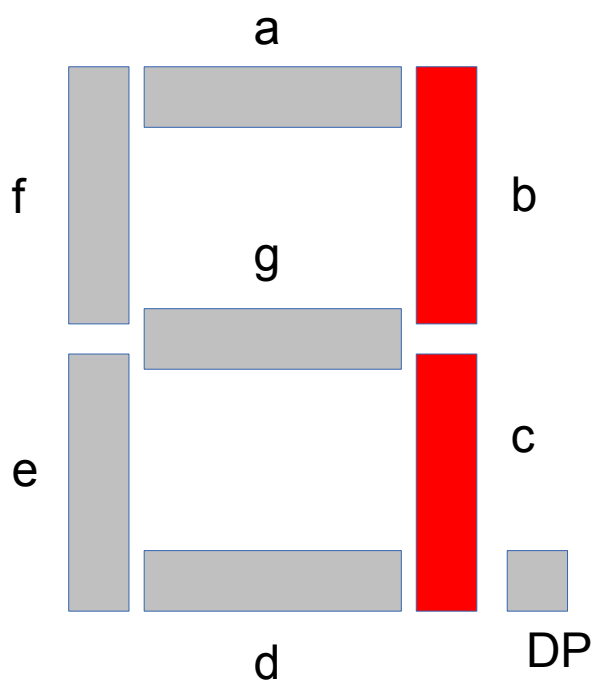


Segment	State
a	ON
b	ON
c	ON
d	ON
e	ON
f	ON
g	OFF
DP	OFF

Display the number 0

Using a simple display

- 7 Segment display hardware

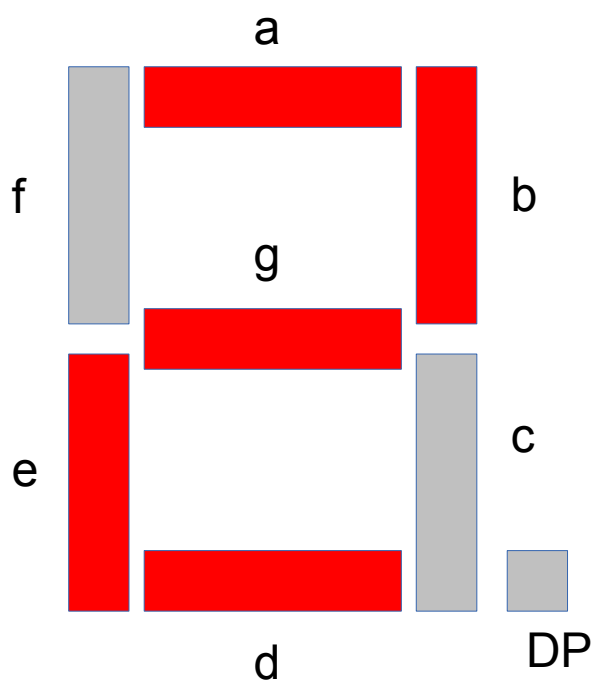


Segment	State
a	OFF
b	ON
c	ON
d	OFF
e	OFF
f	OFF
g	OFF
DP	OFF

Display the number 1

Using a simple display

- 7 Segment display hardware

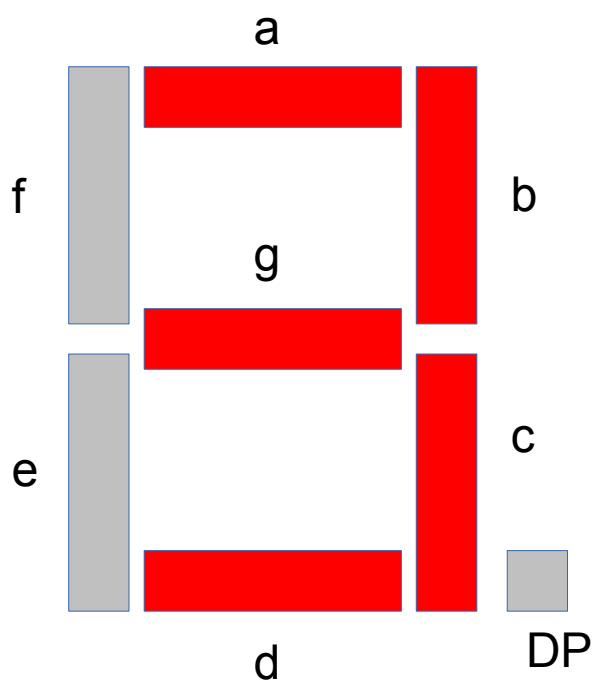


Segment	State
a	ON
b	ON
c	OFF
d	ON
e	ON
f	OFF
g	ON
DP	OFF

Display the number 2

Using a simple display

- 7 Segment display hardware

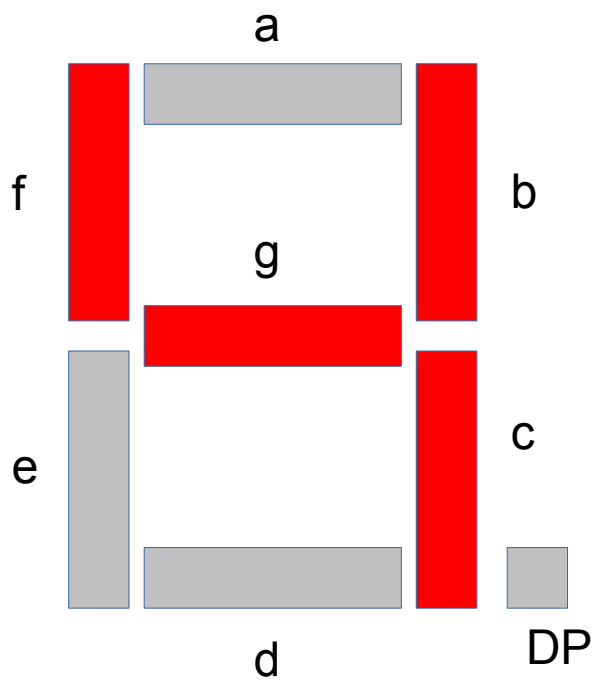


Segment	State
a	ON
b	ON
c	ON
d	ON
e	OFF
f	OFF
g	ON
DP	OFF

Display the number 3

Using a simple display

- 7 Segment display hardware

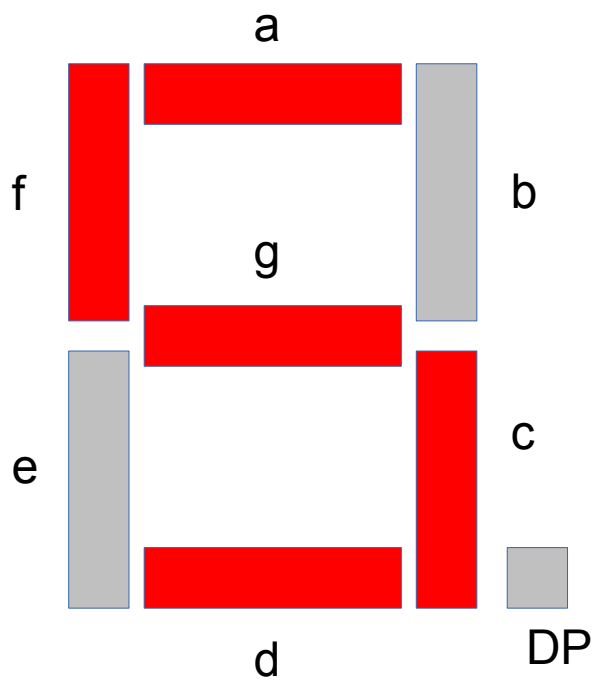


Segment	State
a	OFF
b	ON
c	ON
d	OFF
e	OFF
f	ON
g	ON
DP	OFF

Display the number 4

Using a simple display

- 7 Segment display hardware

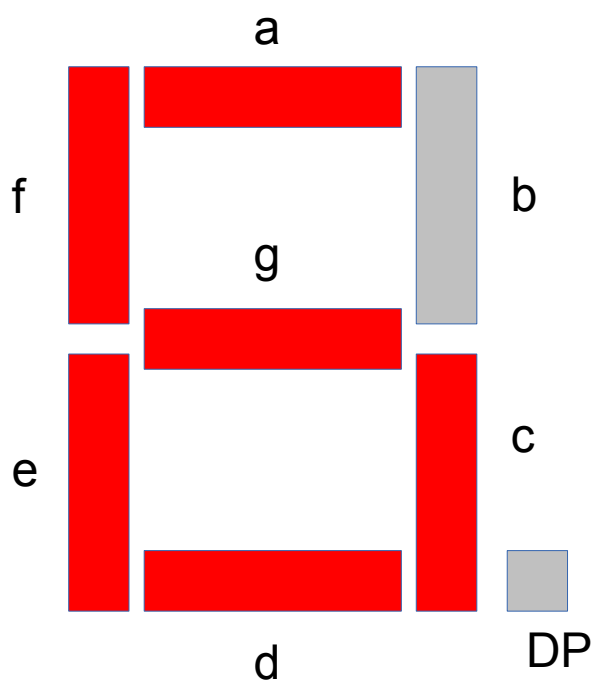


Segment	State
a	ON
b	OFF
c	ON
d	ON
e	OFF
f	ON
g	ON
DP	OFF

Display the number 5

Using a simple display

- 7 Segment display hardware

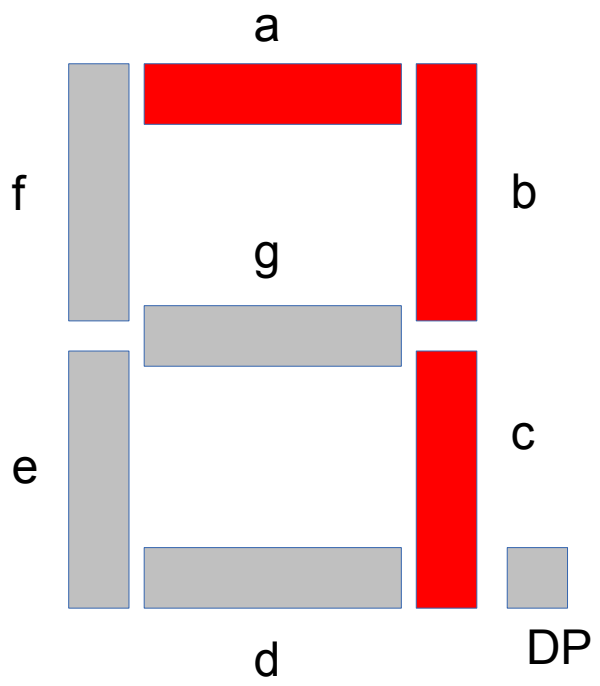


Segment	State
a	ON
b	OFF
c	ON
d	ON
e	ON
f	ON
g	ON
DP	OFF

Display the number 6

Using a simple display

- 7 Segment display hardware

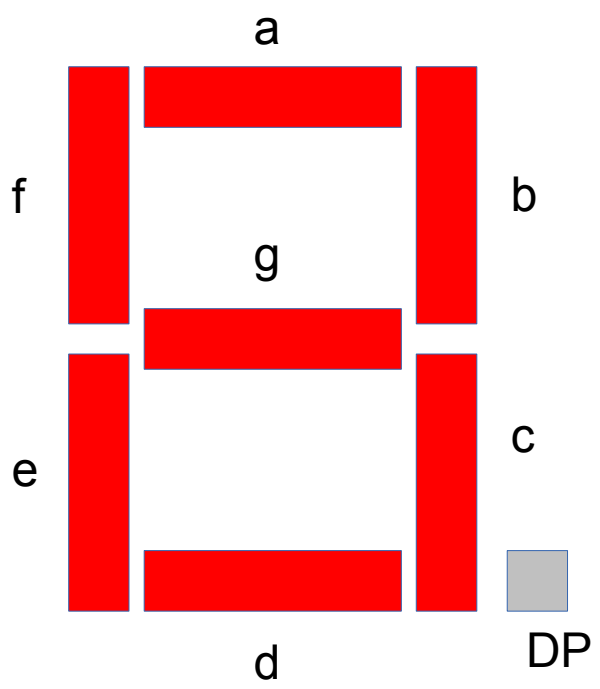


Segment	State
a	ON
b	ON
c	ON
d	OFF
e	OFF
f	OFF
g	OFF
DP	OFF

Display the number 7

Using a simple display

- 7 Segment display hardware

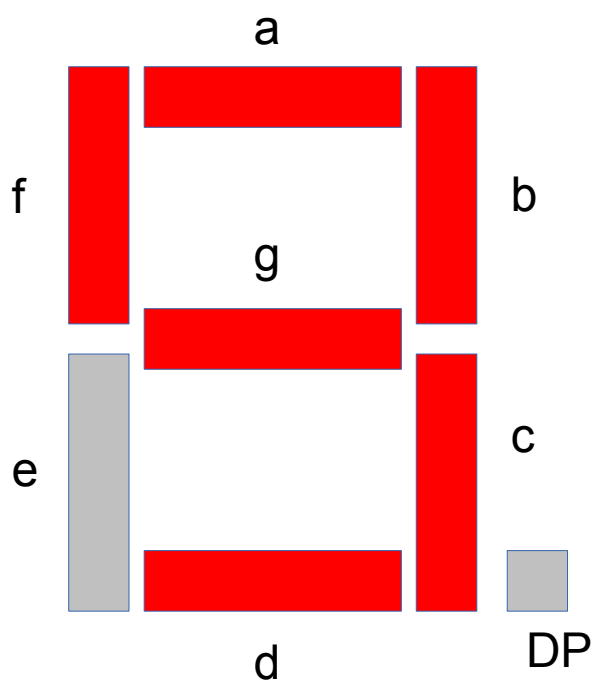


Segment	State
a	ON
b	ON
c	ON
d	ON
e	ON
f	ON
g	ON
DP	OFF

Display the number 8

Using a simple display

- 7 Segment display hardware

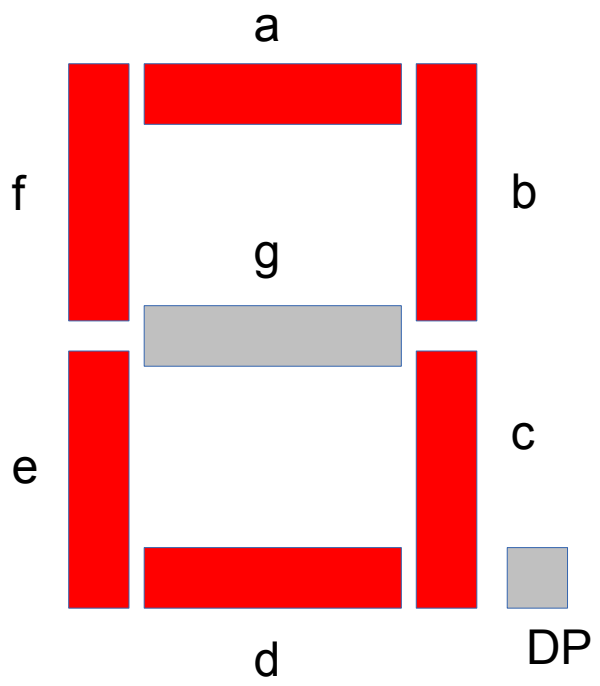


Segment	State
a	ON
b	ON
c	ON
d	ON
e	OFF
f	ON
g	ON
DP	OFF

Display the number 9

Using a simple display

- 7 Segment display hardware

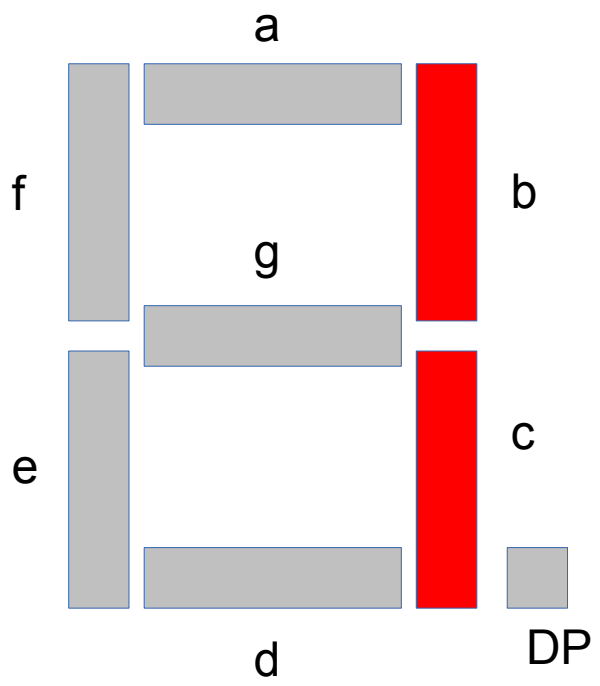


Segment	State
a	1
b	1
c	1
d	1
e	1
f	1
g	0
DP	0

Display the number 0;
Bit pattern:
0011 1111 in binary
0x3 f in hex

Using a simple display

- 7 Segment display hardware

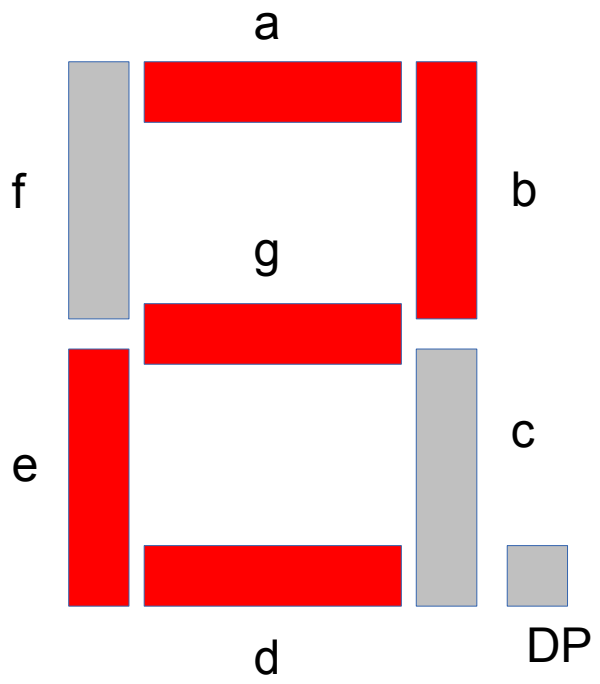


Segment	State
a	0
b	1
c	1
d	0
e	0
f	0
g	0
DP	0

Display the number 1;
Bit pattern:
0000 0110 in binary
0x0 6 in hex

Using a simple display

- 7 Segment display hardware

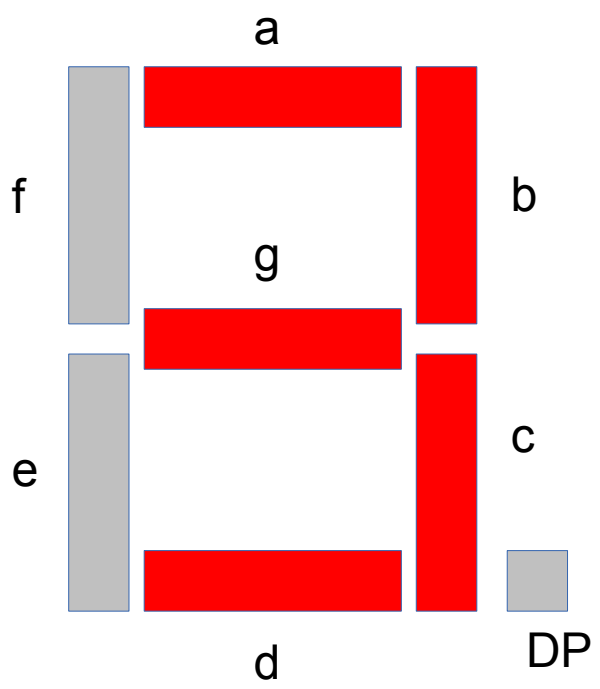


Segment	State
a	1
b	1
c	0
d	1
e	1
f	0
g	1
DP	0

Display the number 2;
Bit pattern:
0101 1011 in binary
0x5 b in hex

Using a simple display

- 7 Segment display hardware

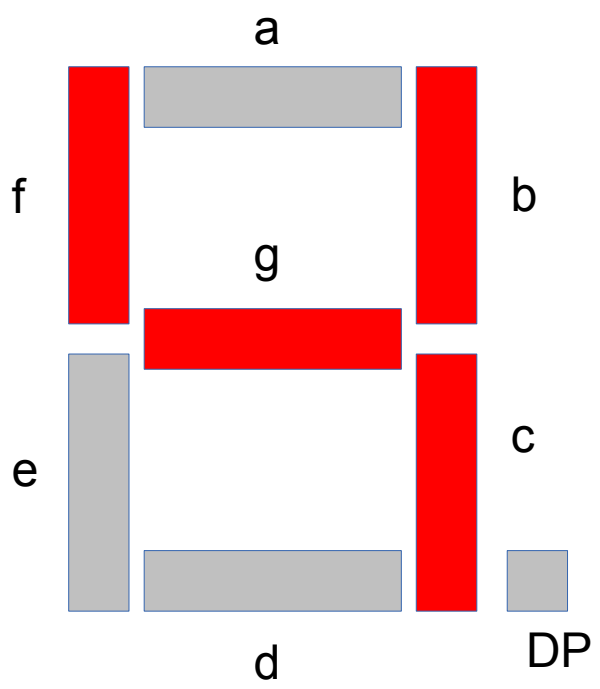


Segment	State
a	1
b	1
c	1
d	1
e	0
f	0
g	1
DP	0

Display the number 3;
Bit pattern:
0100 1111 in binary
0x4 f in hex

Using a simple display

- 7 Segment display hardware

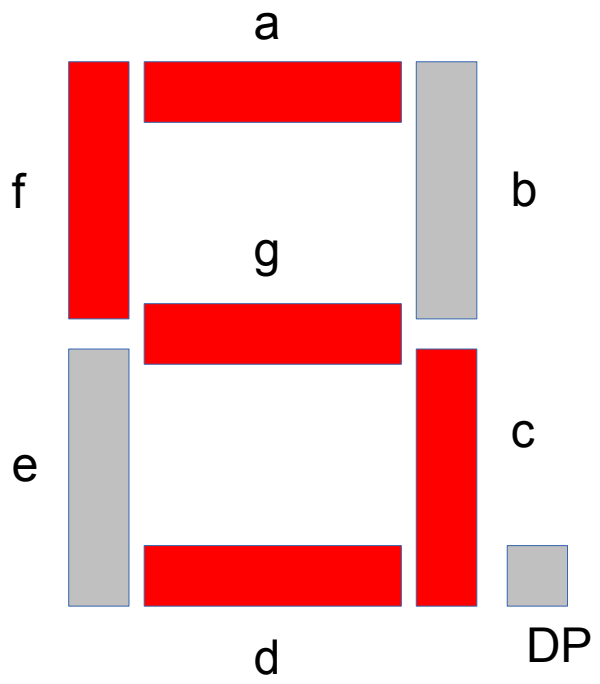


Segment	State
a	0
b	1
c	1
d	0
e	0
f	1
g	1
DP	0

Display the number 4;
Bit pattern:
0110 0110 in binary
0x6 6 in hex

Using a simple display

- 7 Segment display hardware

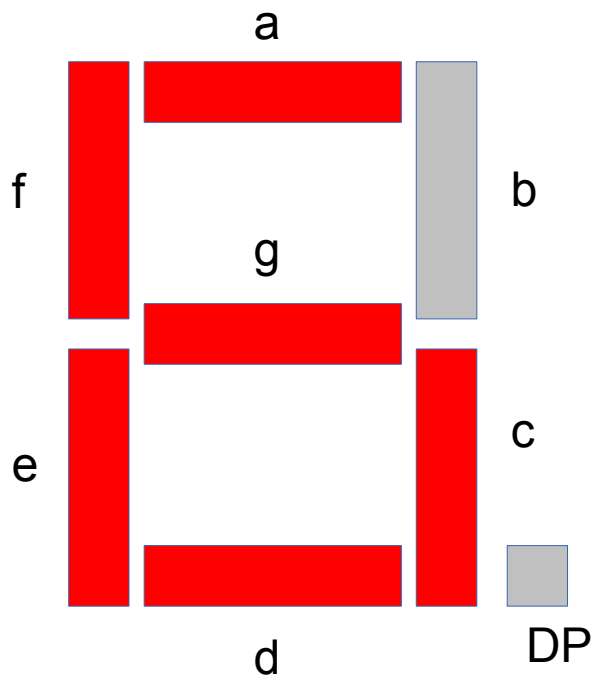


Segment	State
a	1
b	0
c	1
d	1
e	0
f	1
g	1
DP	0

Display the number 5;
Bit pattern:
0110 1101 in binary
0x6 d in hex

Using a simple display

- 7 Segment display hardware

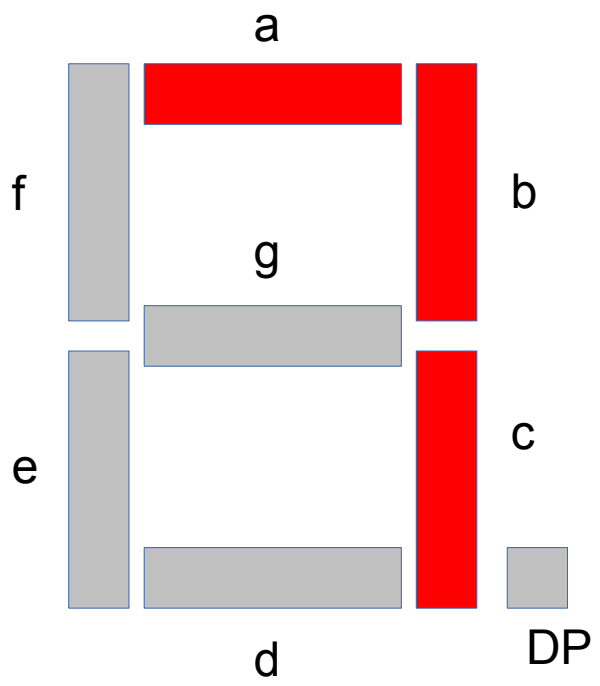


Segment	State
a	1
b	0
c	1
d	1
e	1
f	1
g	1
DP	0

Display the number 6;
Bit pattern:
0111 1101 in binary
0x7 d in hex

Using a simple display

- 7 Segment display hardware

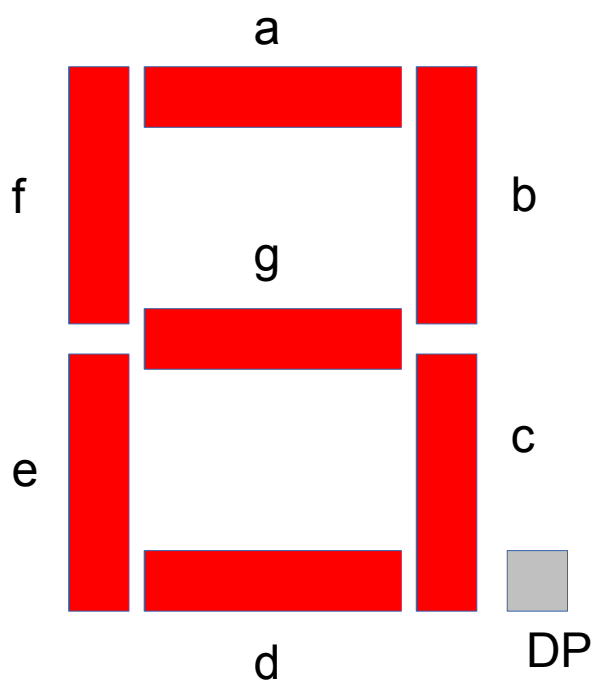


Segment	State
a	1
b	1
c	1
d	0
e	0
f	0
g	0
DP	0

Display the number 7;
Bit pattern:
0000 0111 in binary
0x0 7 in hex

Using a simple display

- 7 Segment display hardware

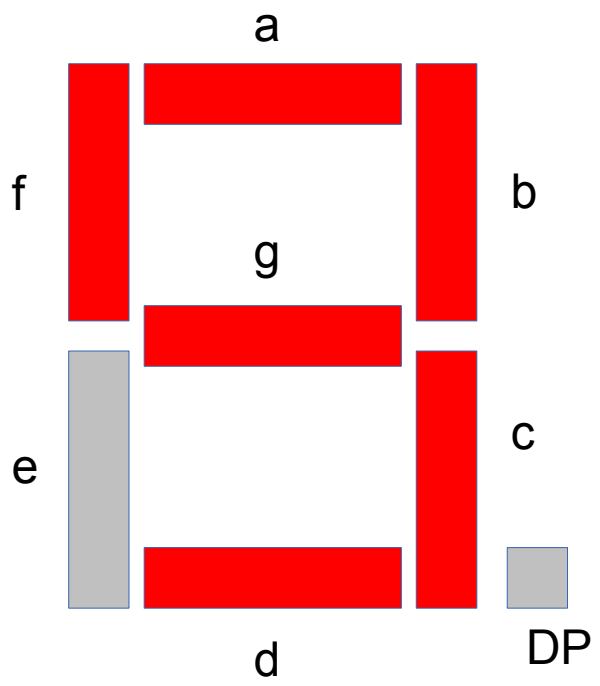


Segment	State
a	1
b	1
c	1
d	1
e	1
f	1
g	1
DP	0

Display the number 8;
Bit pattern:
0111 1111 in binary
0x7 f in hex

Using a simple display

- 7 Segment display hardware



Segment	State
a	1
b	1
c	1
d	1
e	0
f	1
g	1
DP	0

Display the number 3;
Bit pattern:
0110 1111 in binary
0x6 f in hex

Using a simple display

- We can perform translation using a “lookup” table:

```
const char LUT[] =
{0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};
```

- Test: What is LUT[4]?

```
{0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};
```

Using a simple display

- Translation function:

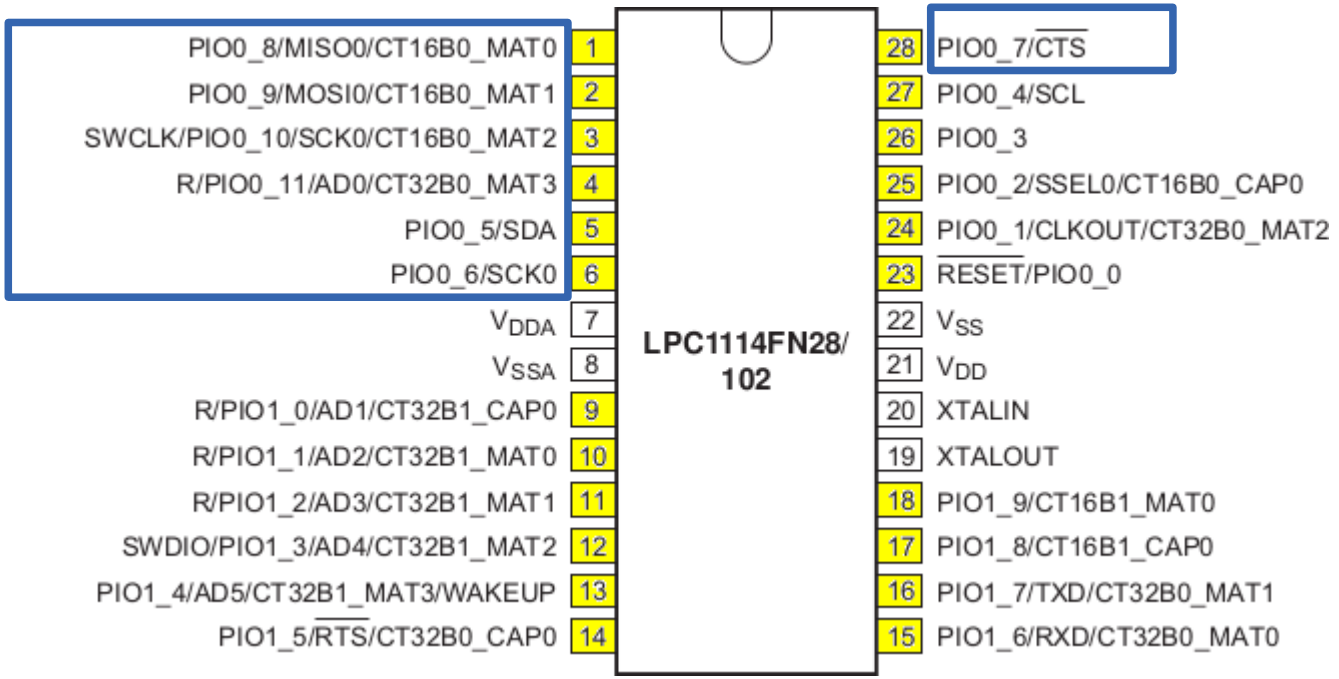
```
const char LUT[] =
{0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};


char DigitToCode(int digit)
{
   if ( (digit >=0) && (digit < 10) )
      return LUT[digit];
   else
      return 0;
}
```

Using a simple display

- Wiring a display
  - Identify suitable “handy” pins on microcontroller
  - Select pins that are “bitwise contiguous” if possible

Using a simple display



Selecting bits GPIO0_5 to GPIO0_11
Total 7 bits
Not driving Decimal point

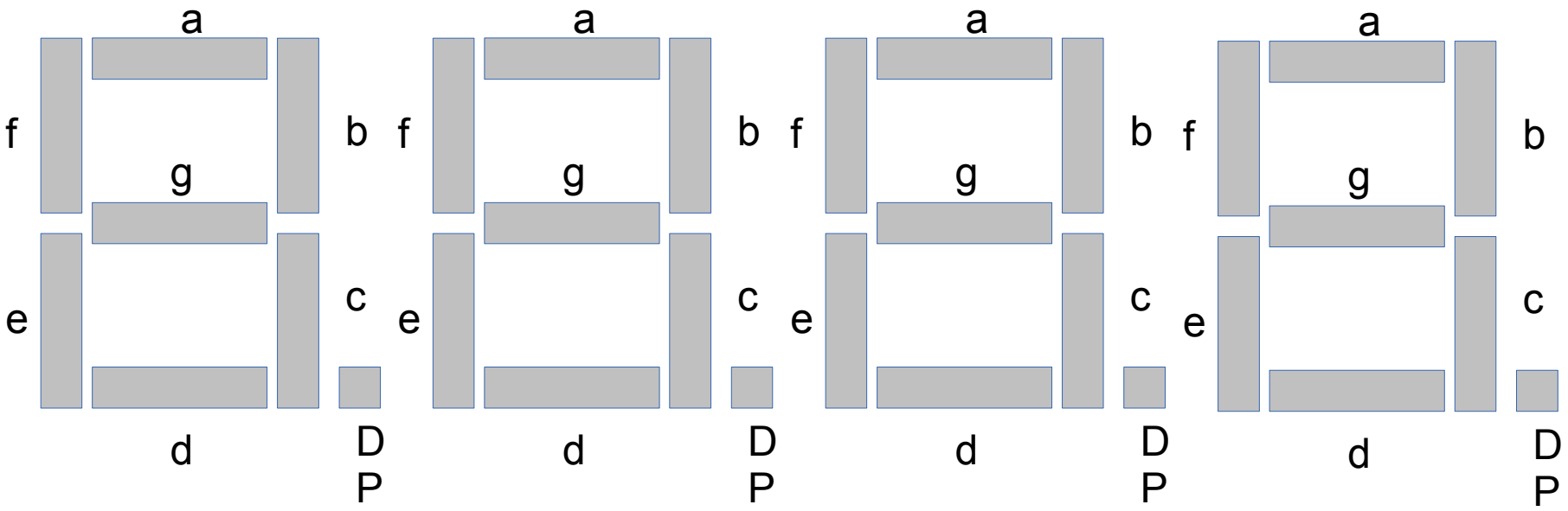
Using a simple display

- Display digit function

```
displayDigit(int Digit)
{
    char code;
    code=Digit2Code(Digit);
    GPIO0DATA = (code << 5);
}
```

Using a simple display

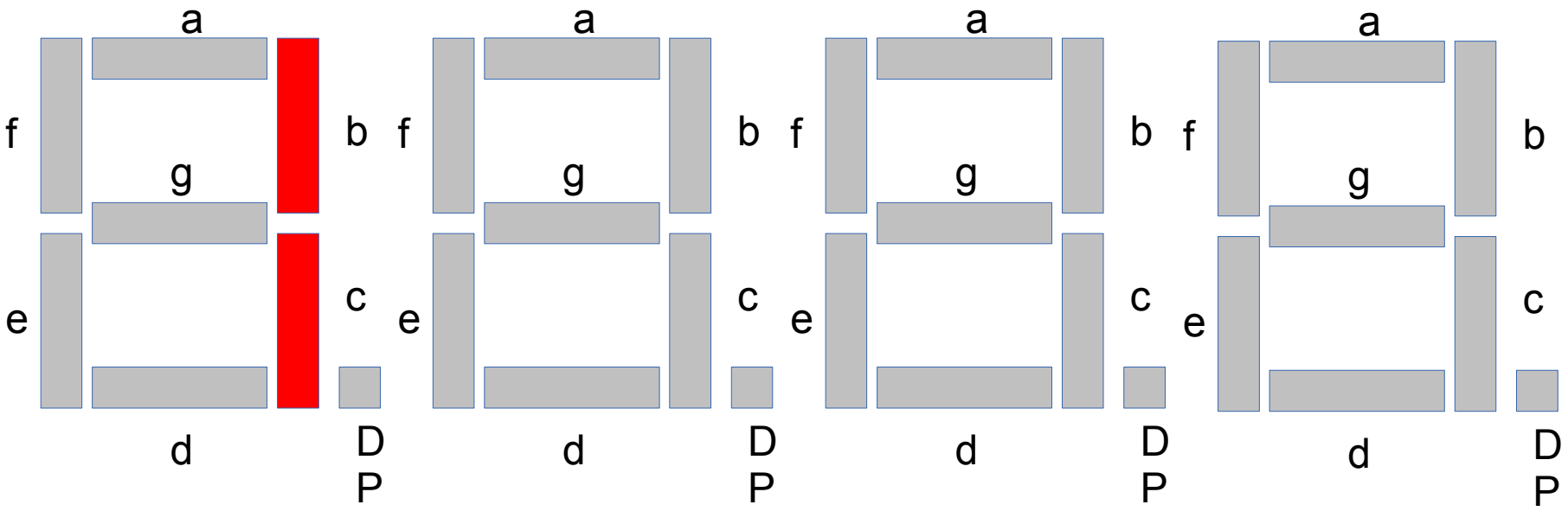
- Most display have more than one digit



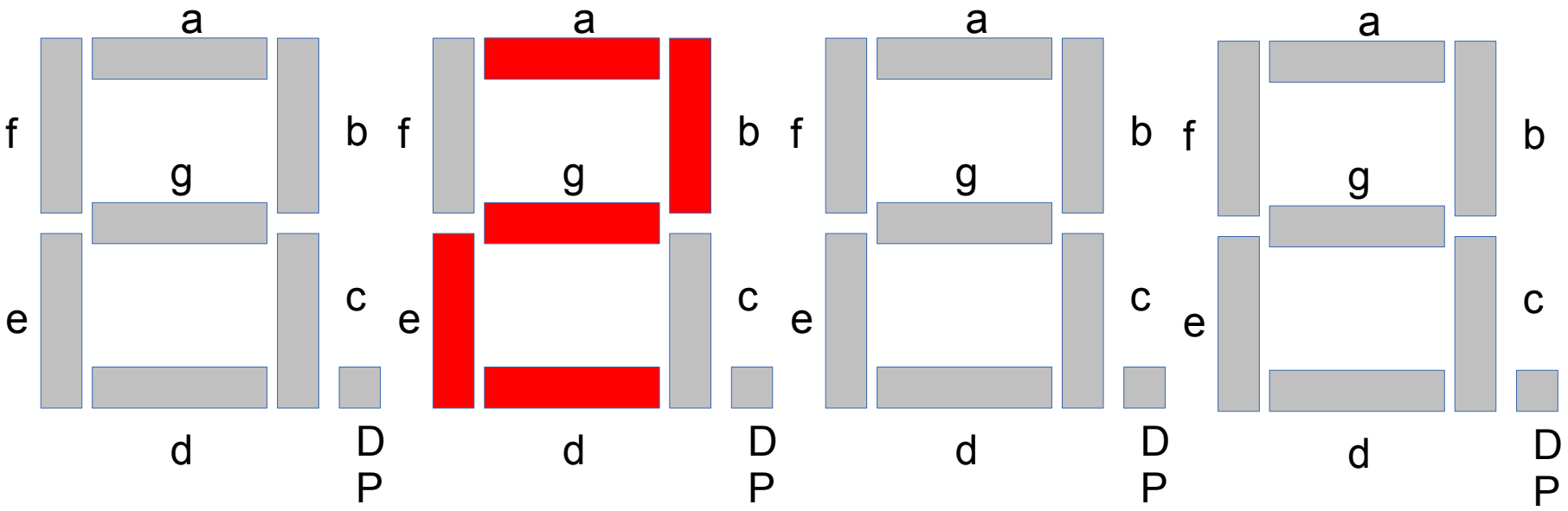
Using a simple display

- Can wire each segment to an individual port pin
- Requires 4x8 pins = 32 Output pins
- Not really necessary
- Rarely done
- Muliplex instead!

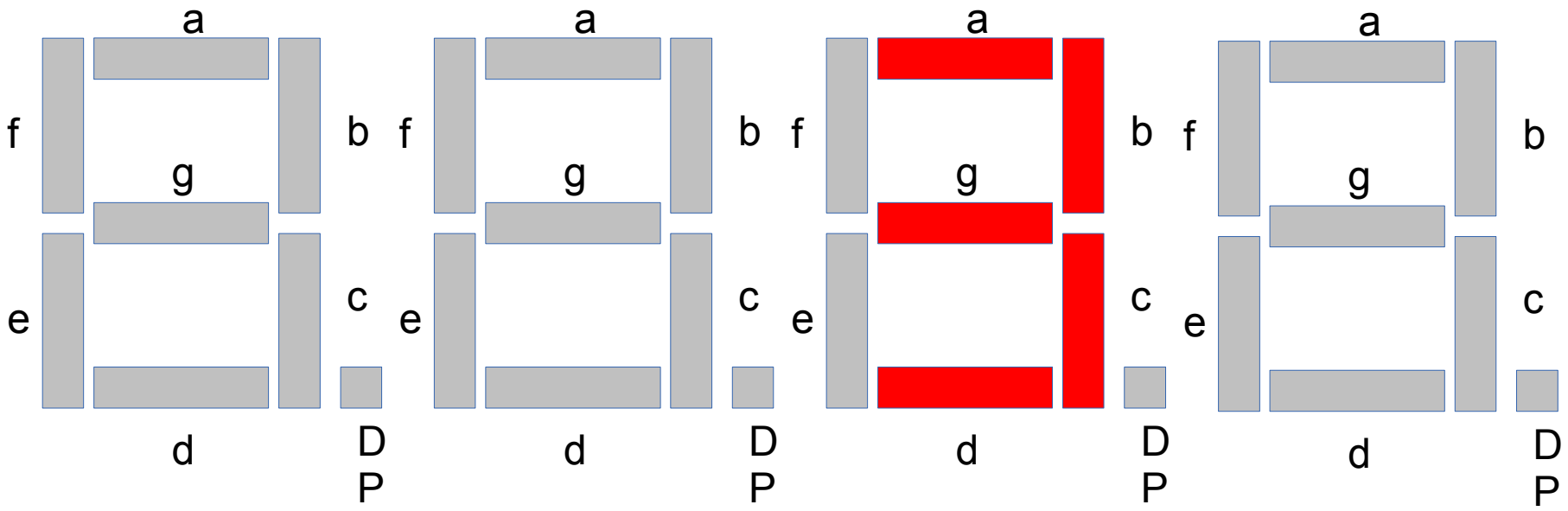
Using a simple display



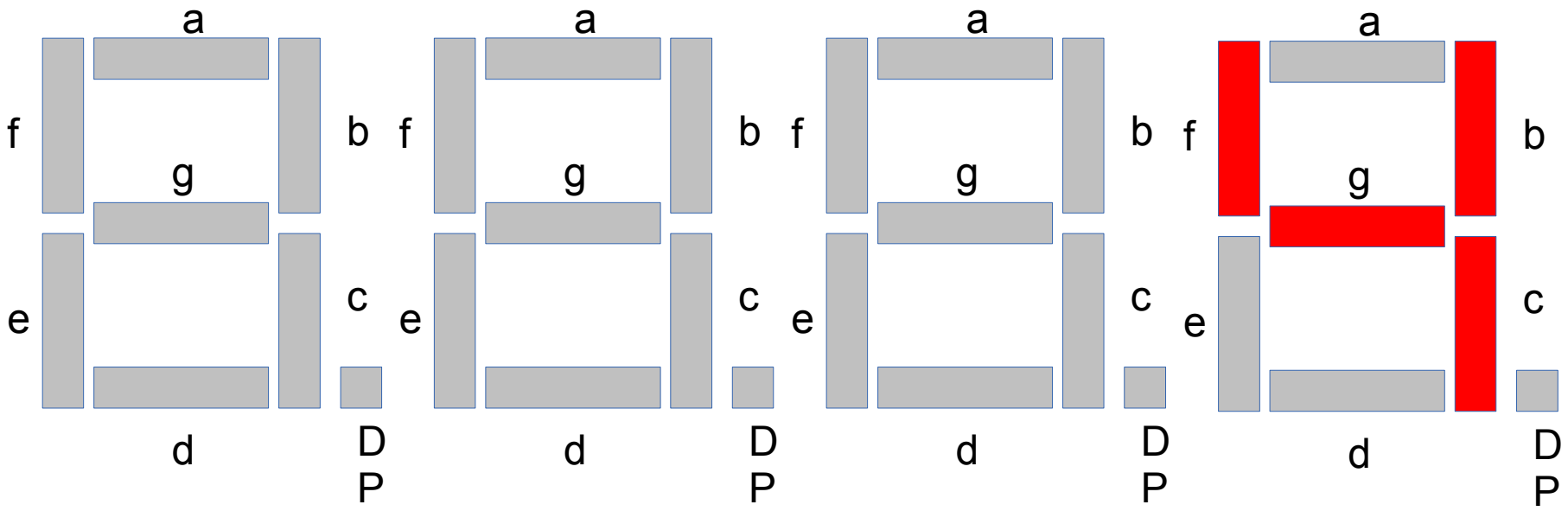
Using a simple display



Using a simple display



Using a simple display

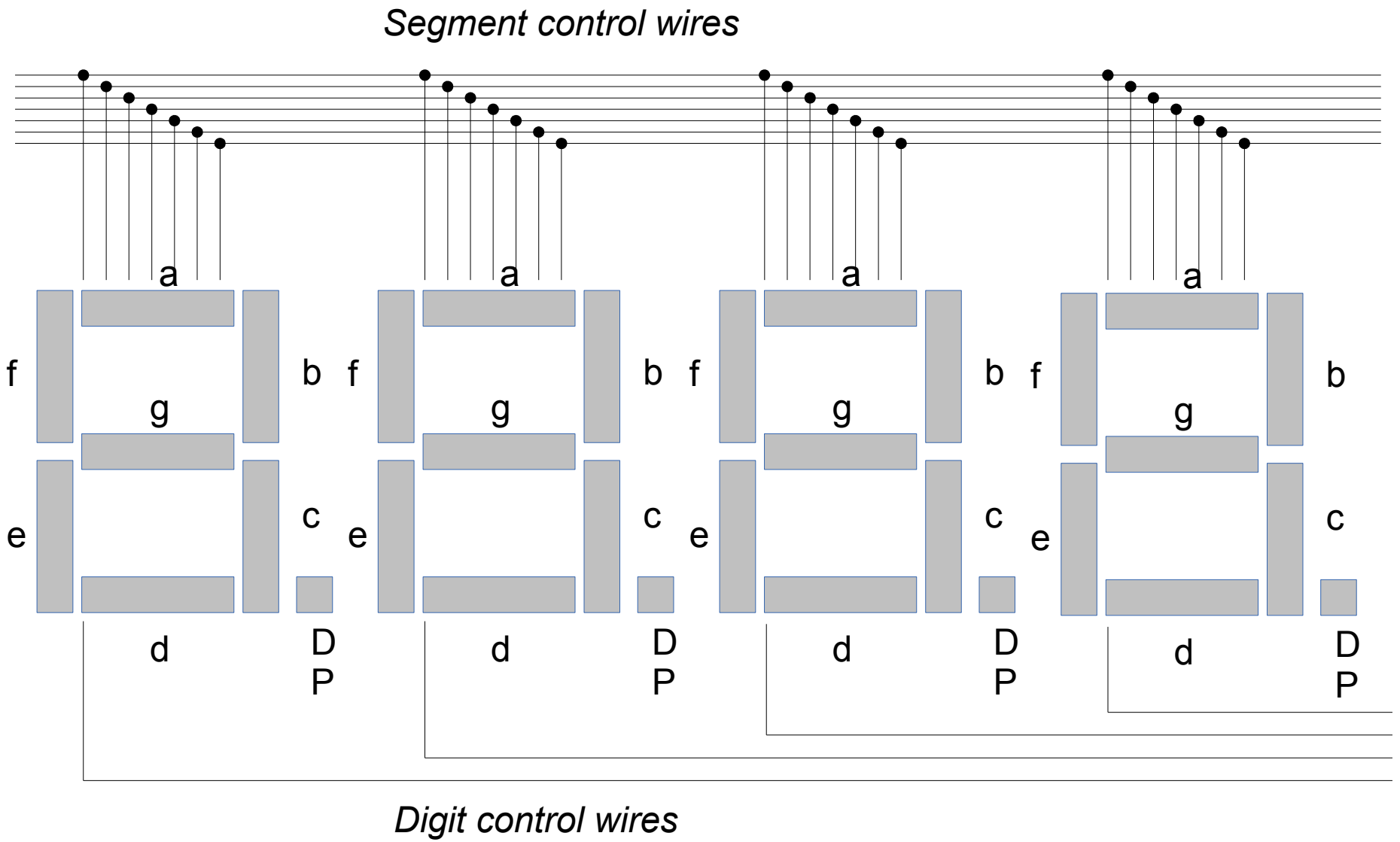


Using a simple display

- Digits are flashed in sequence
- Sequence runs quickly enough to fool the eye
- Pins required:
  - 1 x 8 + 1 for each digit.

Using a simple display

- Most display have more than one digit

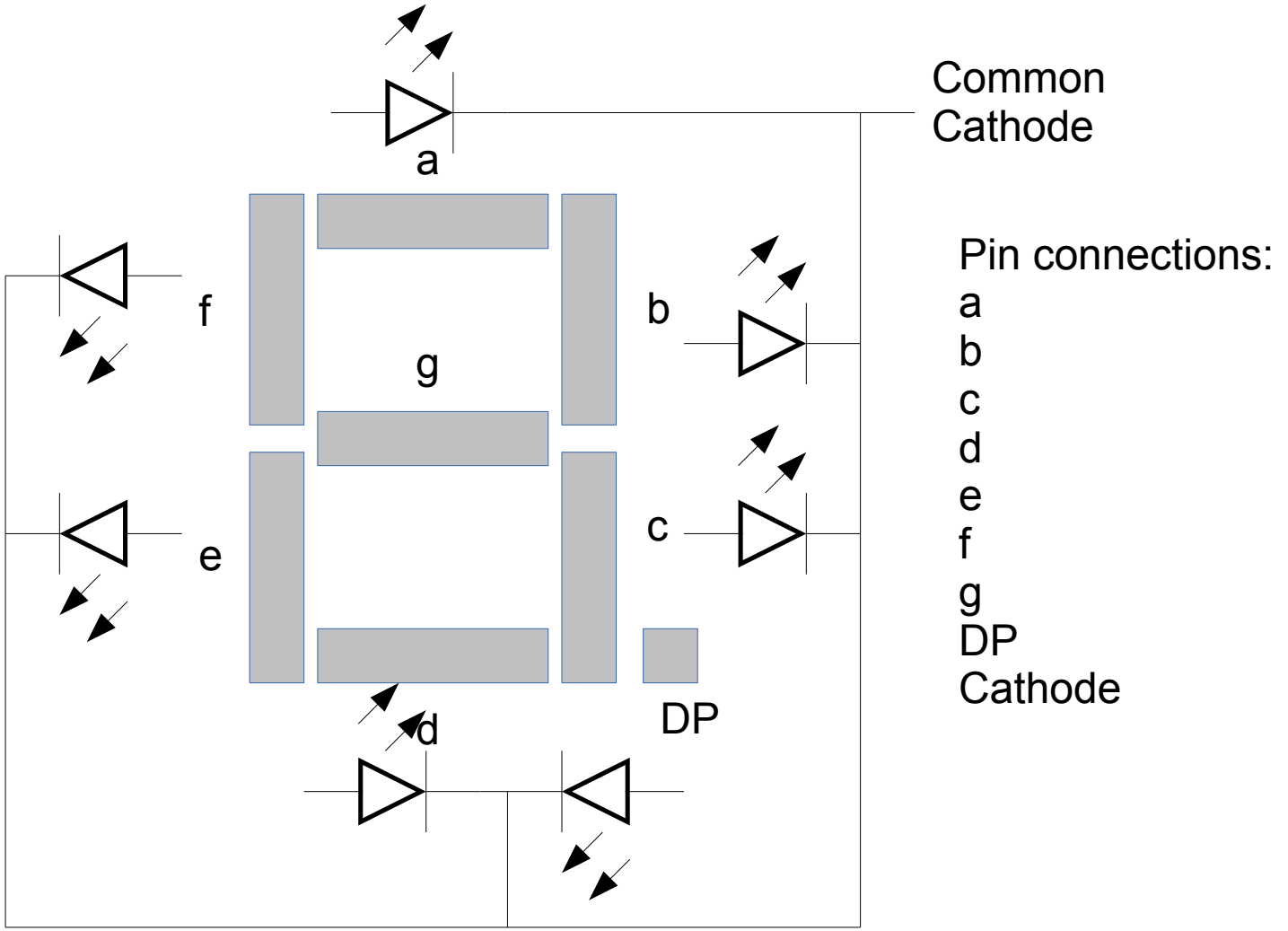


Using a simple display

- Software sequence:
- Get first digit value
- Translate to segment code
- Enable first digit
- Output code
- Pause
- Disable digit
- Repeat for all other digits
- Repeat entire sequence

Using a simple display

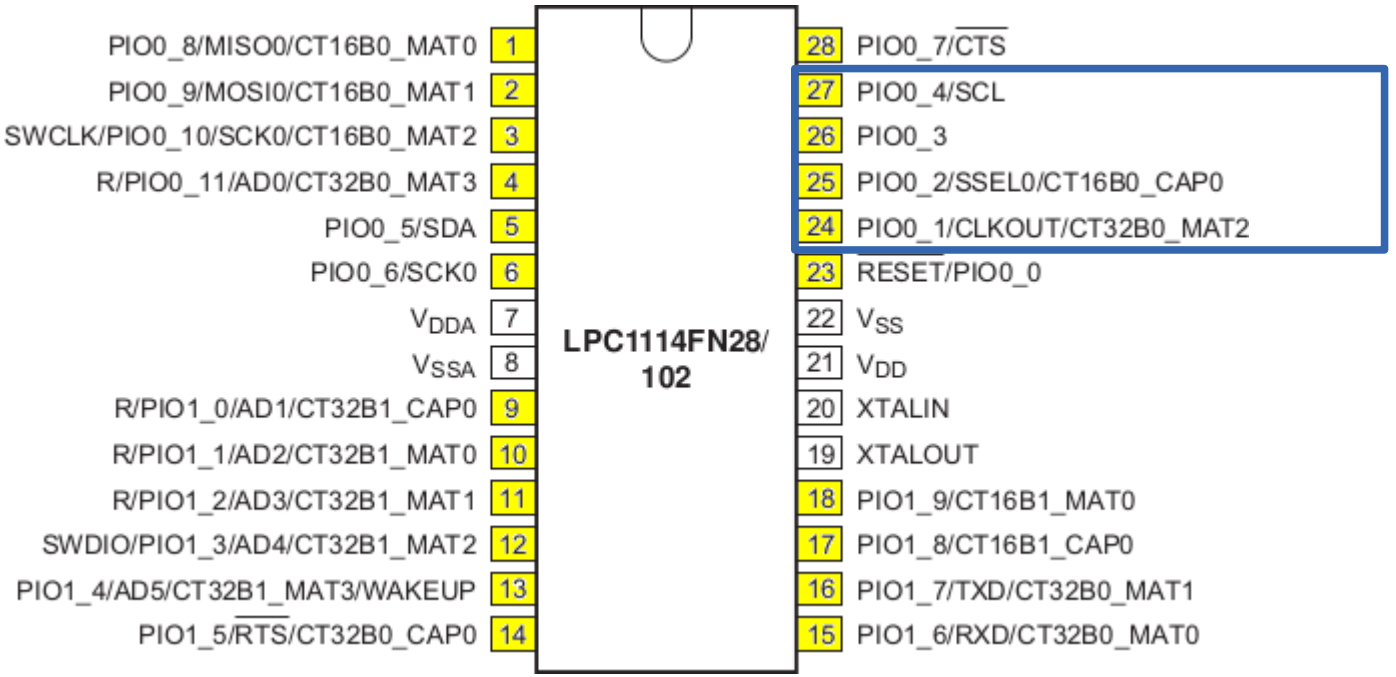
- Internal wiring of 7 segment displays (common cathode)



Using a simple display

- To disable digit, make common cathode high
- To enable digit, make common cathode low
- Can use this for multiplexing
- Must assign additional IO to digit control

Using a simple display



Will use GPIO1 to GPIO4 for digit control.
Pins connected to common cathodes on 4 digit LED display.
High disables a digit