RPN Calculator

Feifei Kong
Nicole Lewis
Vanshil Shah

HP 20b Business Calculator

Finance

Insurance

Real Estate

Accounting

Statistics

PROGRAMMING

User Guide

Equation:
2+5

Stack

Equation:
2+5

Stack

Equation:
2+5

Stack

2

Equation:
2+5

Stack

2

Equation:
2+5

Stack

2
5

Equation:
2+5

Stack

2
7

Equation:
(5+3)x(11-4)

Stack

Equation:
(5+3)x(11-4)

Stack

Equation:
(5+3)x(11-4)

Stack

5

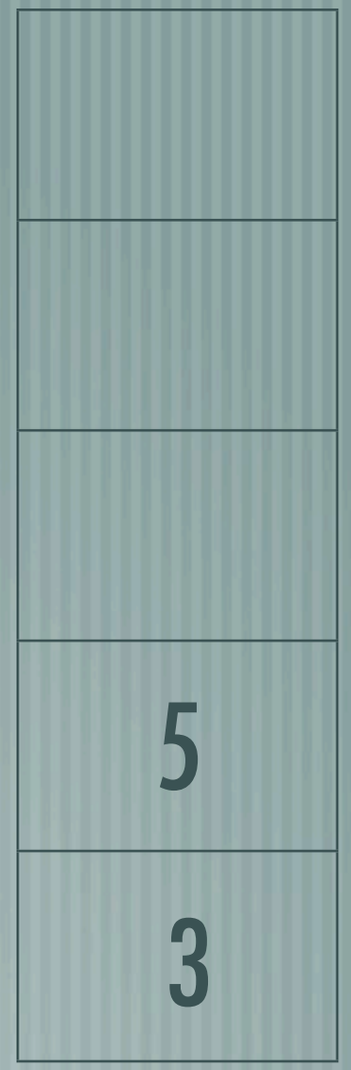
Equation:
(5+3)x(11-4)

Stack

5

Equation:
(5+3)x(11-4)

Stack



Equation:
(5+3)x(11-4)

Stack

5
8

Equation:
(5+3)x(11-4)

Stack

5
8

Equation:
(5+3)x(11-4)

Stack

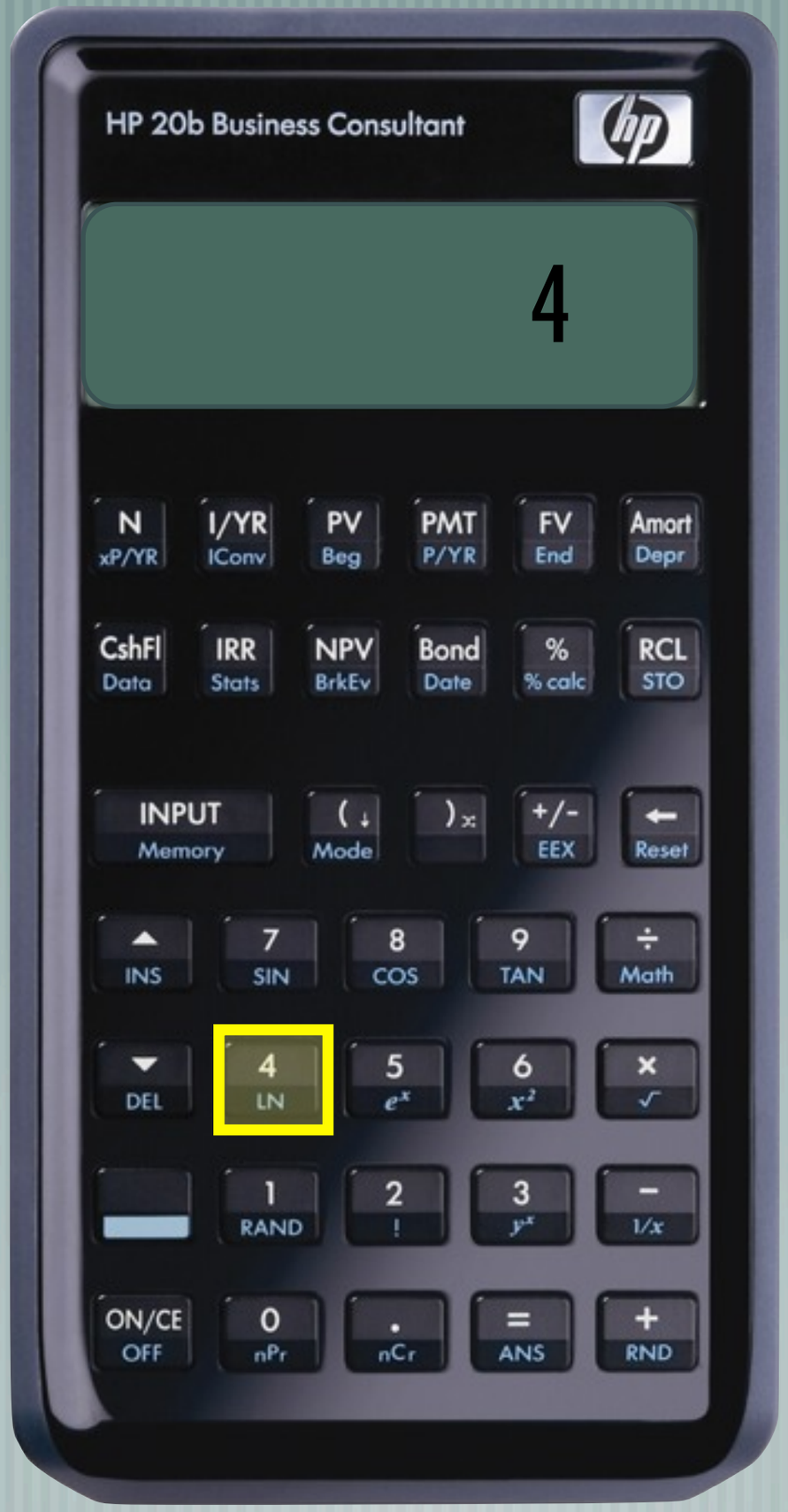
5
8

Equation:
(5+3)x(11-4)

Stack

8
11

Equation:
(5+3)x(11-4)



Stack

8
11

Equation:
(5+3)x(11-4)

Stack

8
8
7

Equation:
(5+3)x(11-4)

Stack

8
8
54

Other Keys


HP 20b Business Consultant
N
xP/YR
I/YR
IConv
PV
Beg
PMT
P/YR
FV
End
Amort
Depr
CshFl
Data
IRR
Stats
NPV
BrkEv
Bond
Date
%
% calc
RCL
STO
INPUT
Memory
(↓
Mode
) x
+/-
EEX
Reset
INS
7
SIN
8
COS
9
TAN
÷
Math
DEL
4
LN
5
e^x
6
x^2
×
$\sqrt{}$
1
RAND
2
!
3
y^x
−
$1/x$
ON/CE
OFF
0
nPr
.
nCr
=
ANS
+
RND

Other Keys


HP 20b Business Consultant
0
N
xP/YR
I/YR
IConv
PV
Beg
PMT
P/YR
FV
End
Amort
Depr
CshFl
Data
IRR
Stats
NPV
BrkEv
Bond
Date
%
% calc
RCL
STO
INPUT
Memory
(↓
Mode
) x
+/-
EEX
Reset
INS
7
SIN
8
COS
9
TAN
÷
Math
DEL
4
LN
5
e^x
6
x^2
×
√
1
RAND
2
!
3
y^x
−
1/x
ON/CE
OFF
0
nPr
.
nCr
=
ANS
+
RND

Other Keys


HP 20b Business Consultant
- 0
N
xP/YR
I/YR
IConv
PV
Beg
PMT
P/YR
FV
End
Amort
Depr
CshFl
Data
IRR
Stats
NPV
BrkEv
Bond
Date
%
% calc
RCL
STO
INPUT
Memory
(↓
Mode
) x
+/-
EEX
Reset
INS
7
SIN
8
COS
9
TAN
÷
Math
DEL
4
LN
5
e^x
6
x^2
×
$\sqrt{}$
1
RAND
2
!
3
y^x
−
$1/x$
ON/CE
OFF
0
nPr
.
nCr
=
ANS
+
RND

Other Keys


HP 20b Business Consultant
0
N
xP/YR
I/YR
IConv
PV
Beg
PMT
P/YR
FV
End
Amort
Depr
CshFl
Data
IRR
Stats
NPV
BrkEv
Bond
Date
%
% calc
RCL
STO
INPUT
Memory
(↓
Mode
) x
+/-
EEX
Reset
INS
7
SIN
8
COS
9
TAN
÷
Math
DEL
4
LN
5
e^x
6
x^2
×
$\sqrt{}$
1
RAND
2
!
3
y^x
−
$1/x$
ON/CE
OFF
0
nPr
.
nCr
=
ANS
+
RND

Other Keys


HP 20b Business Consultant
0
N
xP/YR
I/YR
IConv
PV
Beg
PMT
P/YR
FV
End
Amort
Depr
CshFl
Data
IRR
Stats
NPV
BrkEv
Bond
Date
%
% calc
RCL
STO
INPUT
Memory
(↓
Mode
) x
+/-
EEX
Reset
INS
7
SIN
8
COS
9
TAN
÷
Math
DEL
4
LN
5
e^x
6
x^2
×
$\sqrt{}$
1
RAND
2
!
3
y^x
−
$1/x$
ON/CE
OFF
0
nPr
.
nCr
=
ANS
+
RND

Other Keys

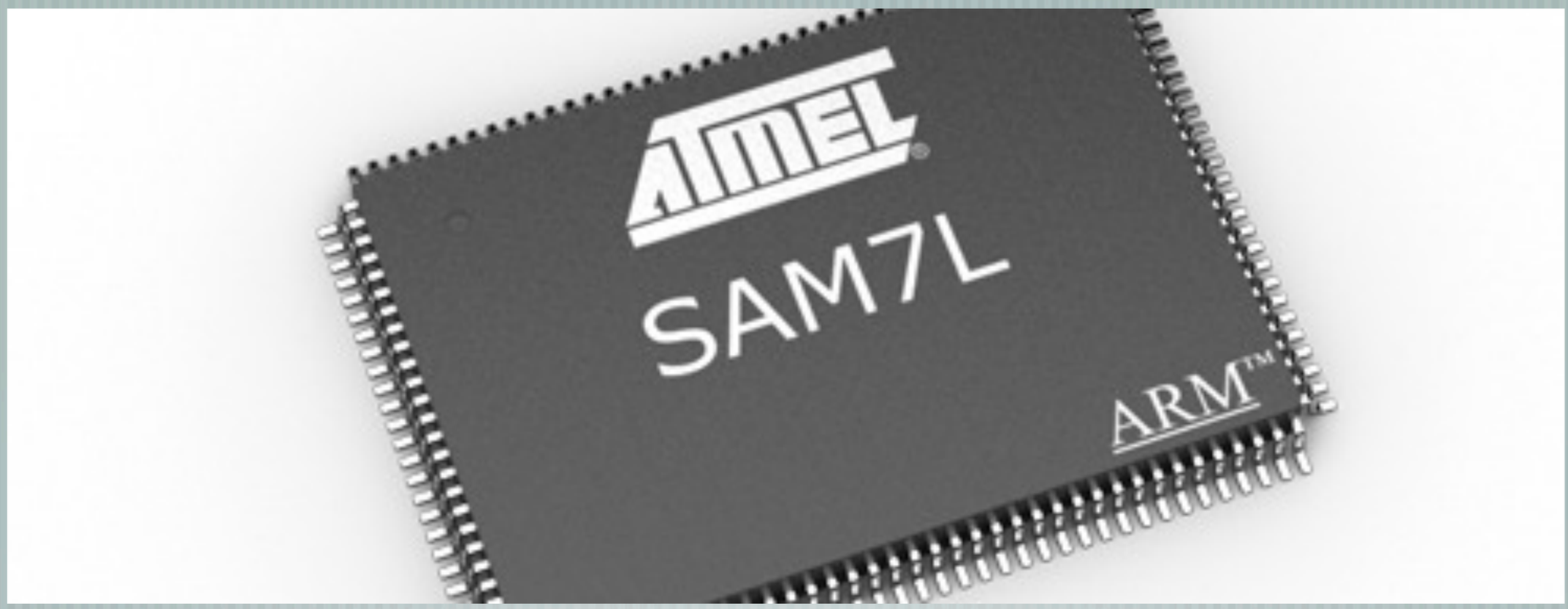

HP 20b Business Consultant
0
N
xP/YR
I/YR
IConv
PV
Beg
PMT
P/YR
FV
End
Amort
Depr
CshFl
Data
IRR
Stats
NPV
BrkEv
Bond
Date
%
% calc
RCL
STO
INPUT
Memory
(↓
Mode
) x
+/-
EEX
Reset
INS
7
SIN
8
COS
9
TAN
÷
Math
DEL
4
LN
5
e^x
6
x^2
×
√
1
RAND
2
!
3
y^x
−
1/x
ON/CE
OFF
0
nPr
.
nCr
=
ANS
+
RND

Other Keys

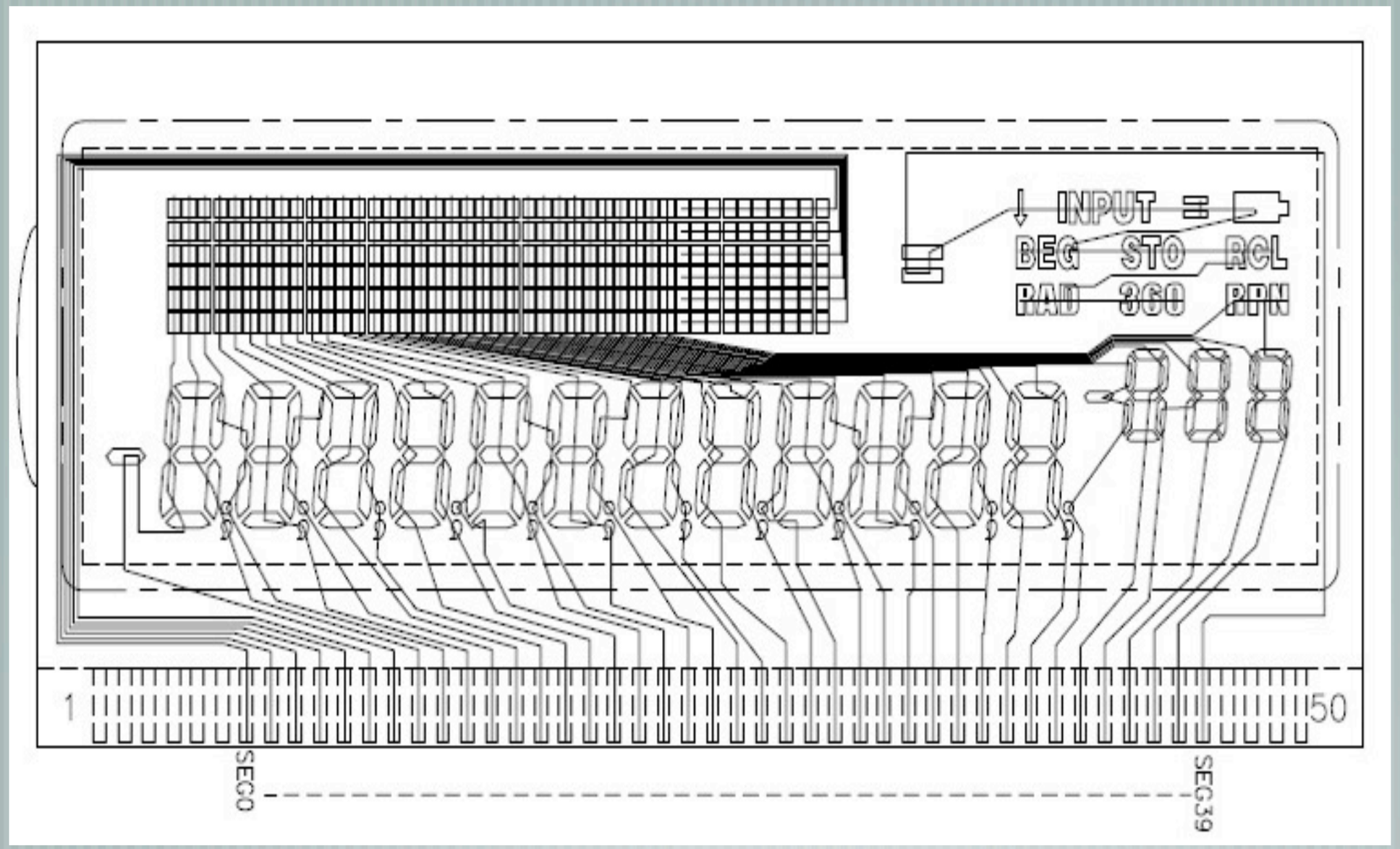

HP 20b Business Consultant
error
N
xP/YR
I/YR
IConv
PV
Beg
PMT
P/YR
FV
End
Amort
Depr
CshFl
Data
IRR
Stats
NPV
BrkEv
Bond
Date
%
% calc
RCL
STO
INPUT
Memory
(↓
Mode
) x
+/-
EEX
Reset
INS
7
SIN
8
COS
9
TAN
÷
Math
DEL
4
LN
5
e^x
6
x^2
×
$\sqrt{}$
1
RAND
2
!
3
y^x
−
$1/x$
ON/CE
OFF
0
nPr
.
nCr
=
ANS
+
RND

Platform

Processor

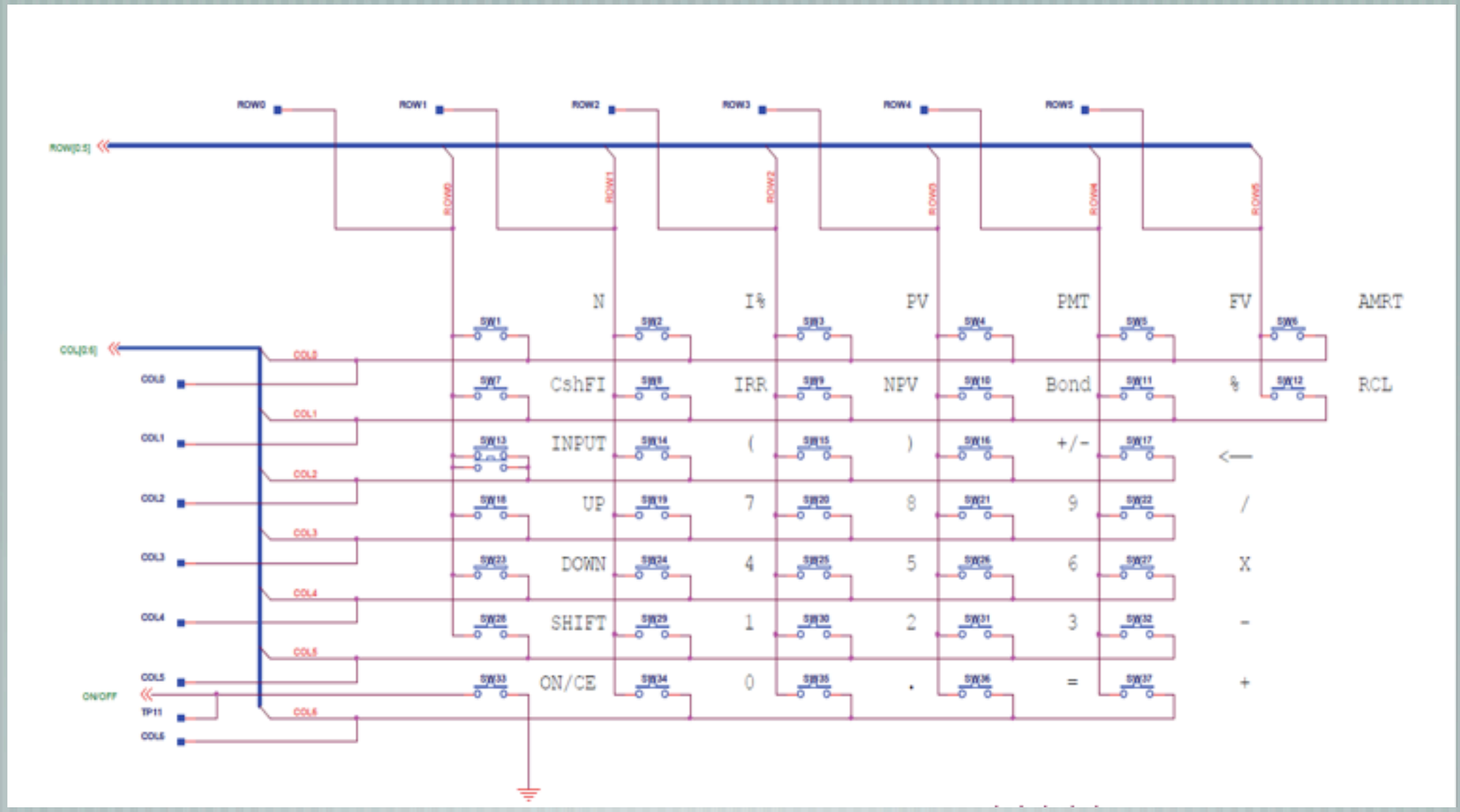


LCD




```
aaa
f   b
f   b
 ggg
e   c
e   c
 ddd
```

The Keyboard



Code

Lab 1 - A Scrolling Display

```
#include "AT91SAM7L128.h"
#include "lcd.h"

int main() {
    lcd_init();
    int i, j, n;
    char a[] = ("Empire State of Mind ");
    for (j = 0; j < 1000; j++) {
        for (i = j; i < j + 21; i++) {
            lcd_put_char7(a[i % 21], i - j);
        }
        for(n = 0; n < 50000; n++) { }
    }
    return 0;
}
```

Lab 2 – Scanning the Keyboard

```
char keyboard_key() {
    char keyb[7][6] = {
            {'N','I', 'P', 'M', 'F', 'A'},
            {'C', 'R', 'V', 'B', '%', 'L'},
            {'U', '(', ')', '_', '<'},
            {'^', '7', '8', '9', '/'},
            {'v', '4', '5', '6', '*'},
            {'S', '1', '2', '3', '-'},
            { 0, '0', '.', '=', '+'}
};
int j, i, a, b;
```

Lab 2 – Scanning the Keyboard

```
for (j = 0; j < 7; j++) {
    keyboard_column_low(j);
    for (i = 0 ; i < 6 ; i++) {
        if (!keyboard_row_read(i)) {
            a = i;
            b = j;
        }
    }
    keyboard_column_high(j);
}
return keyb[b][a];
}
```

Lab 3 – Entering and Displaying Numbers

Numbers:

```
        if (digits < MAX_DIGITS) {
            num = c - '0';
            finalvalue = finalvalue * 10 + num;
        value = abs(finalvalue);
        while (value > 0) {
            disp = (value % 10) + '0';
            lcd_put_char7(disp, LAST_SPOT - count);
            value = value / 10;
            count++;
        }
    }
```

Lab 3 – Entering and Displaying Numbers

+/- Sign:

```
if (c == '~') {
    finalvalue = finalvalue * -1;
    if (neg == 0) {
        lcd_put_char7('-', 0);
        neg = 1;
    }
    else {
        lcd_put_char7(' ', 0);
        neg = 0;
    }
}
```

Lab 3 – Entering and Displaying Numbers

Backspace:

```
        else if (c == '\b') {
            if (digits > 0) {
                finalvalue = finalvalue / 10;
                value = abs(finalvalue);
                while (value > 0) {
                    disp = (value % 10) + '0';
                    lcd_put_char7(disp, LAST_SPOT - count);
                    value = value / 10;
                }
            }
    }
```

Lab 4 – An RPN Calculator

```
if (UserOperation == '+') {
        stack[StackPointer-2] = stack[StackPointer-2] + stack[StackPointer-1];
        StackPointer–;
}
if (UserOperation == '-') {
        stack[StackPointer-2] = stack[StackPointer-2] – stack[StackPointer-1];
        StackPointer–;
}
if(UserOperation == '*') {
        stack[StackPointer-2] = stack[StackPointer-2] * stack[StackPointer-1];
        StackPointer–;
}
```

Lab 4 – An RPN Calculator

```
Value = abs(stack[StackPointer - 1]);
if (value > MAX_NUMBER) {
        lcd_print7("      ERRoR");
        StackPointer=0;
}
else {
        if (value == 0) {
        lcd_put_char7('0', LAST_SPOT);
        lcd_put_char7(' ', LAST_SPOT - 1);
        }
}
if (stack[StackPointer - 1] < 0){
        lcd_put_char7('-', LAST_SPOT - count);
}
```

Conclusions