Design an MT

In [1]:
```
load("bitwise");
```
Out[1]: /usr/local/share/maxima/5.41.0/share/contrib/bitwise/bitwise.lisp

In [2]:
```
seed:matrix([27,17,21,5,30,14,16]);
```
Out[2]:
$$\begin{pmatrix} 27 & 17 & 21 & 5 & 30 & 14 & 16 \end{pmatrix}$$

In [4]:
```
n:matrix_size(seed)[2];
```
Out[4]: 7

In [5]:
```
ident_4x4:ident(4);
```
Out[5]:
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In [6]:
```
zero_4x1:transpose([0,0,0,0]);
```
Out[6]:
$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

In [7]:
```
matrix_A:ident_4x4;
```
Out[7]:
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In [8]:
```
matrix_A:addcol(zero_4x1,matrix_A)$
```
Out[8]:
$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

In [9]:
```
vec_a:[1,0,1,1,0]$
```
Out[9]: $[1,0,1,1,0]$

In [10]:
```
matrix_A:addrow(matrix_A,vec_a)$
```
Out[10]:
$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

In [11]:
```
matrix_A[5]:vec_a;
```
Out[11]: $[1,0,1,1,0]$

In [12]:
```
m:2;
```
Out[12]: 2

In [17]:
```
bit_and(seed[1][m],16);
```
Out[17]: 16

In [18]:
```
bit_rsh(%,4);
```
Out[18]: 1

In [19]:
```
bottom_tap:[ bit_rsh(bit_and(seed[1][m],16),4),
             bit_rsh(bit_and(seed[1][m],8),3),
             bit_rsh(bit_and(seed[1][m],4),2),
             bit_rsh(bit_and(seed[1][m],2),1),
                     bit_and(seed[1][m],1)];
```
Out[19]: $[1,0,0,0,1]$

In [20]:
```
top_tap:[   bit_rsh(bit_and(seed[1][n],16),4),
            bit_rsh(bit_and(seed[1][n-1],8),3),
            bit_rsh(bit_and(seed[1][n-1],4),2),
            bit_rsh(bit_and(seed[1][n-1],2),1),
                    bit_and(seed[1][n-1],1)];
```
Out[20]: $[1,1,1,1,0]$

In [21]:
```
bottom_tap:matrix(bottom_tap);
```
Out[21]: $\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \end{pmatrix}$

In [22]:
```
new_bits:mod(bottom_tap + top_tap  . matrix_A,2);
```
Out[22]: $\begin{pmatrix} 1 & 1 & 1 & 1 & 0 \end{pmatrix}$

In [23]:
```
new_numb:mod(new_bits[1][5]*16+
             new_bits[1][4]*8+
            new_bits[1][3]*4+
            new_bits[1][2]*2+
            new_bits[1][1],32);
```
Out[23]: 15

In [24]:
```
shr7:matrix([0,1,0,0,0,0,0],
             [0,0,1,0,0,0,0],
             [0,0,0,1,0,0,0],
             [0,0,0,0,1,0,0],
             [0,0,0,0,0,1,0],
             [0,0,0,0,0,0,1],
            [0,0,0,0,0,0,0]);
```
Out[24]:
$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

In [25]:
```
seed:seed . shr7;
```
Out[25]: $\begin{pmatrix} 0 & 27 & 17 & 21 & 5 & 30 & 14 \end{pmatrix}$

In [26]:
```
seed[1][1]:new_numb;
```
Out[26]: 15

In [27]:

```
printf(true,"~5,'0b",new_numb);
```

```
01111
```

Out[27]: **false**