```
// PRESENT & LED Sbox
/*****************************************************************************/
/*u4 S[16] = { 0xc, 0x5, 0x6, 0xb, 0x9, 0x0, 0xa, 0xd, 0x3, 0xe, 0xf, 0x8, 0x4, 0x7, 0x1, 0x2,  };
/*****************************************************************************/
```

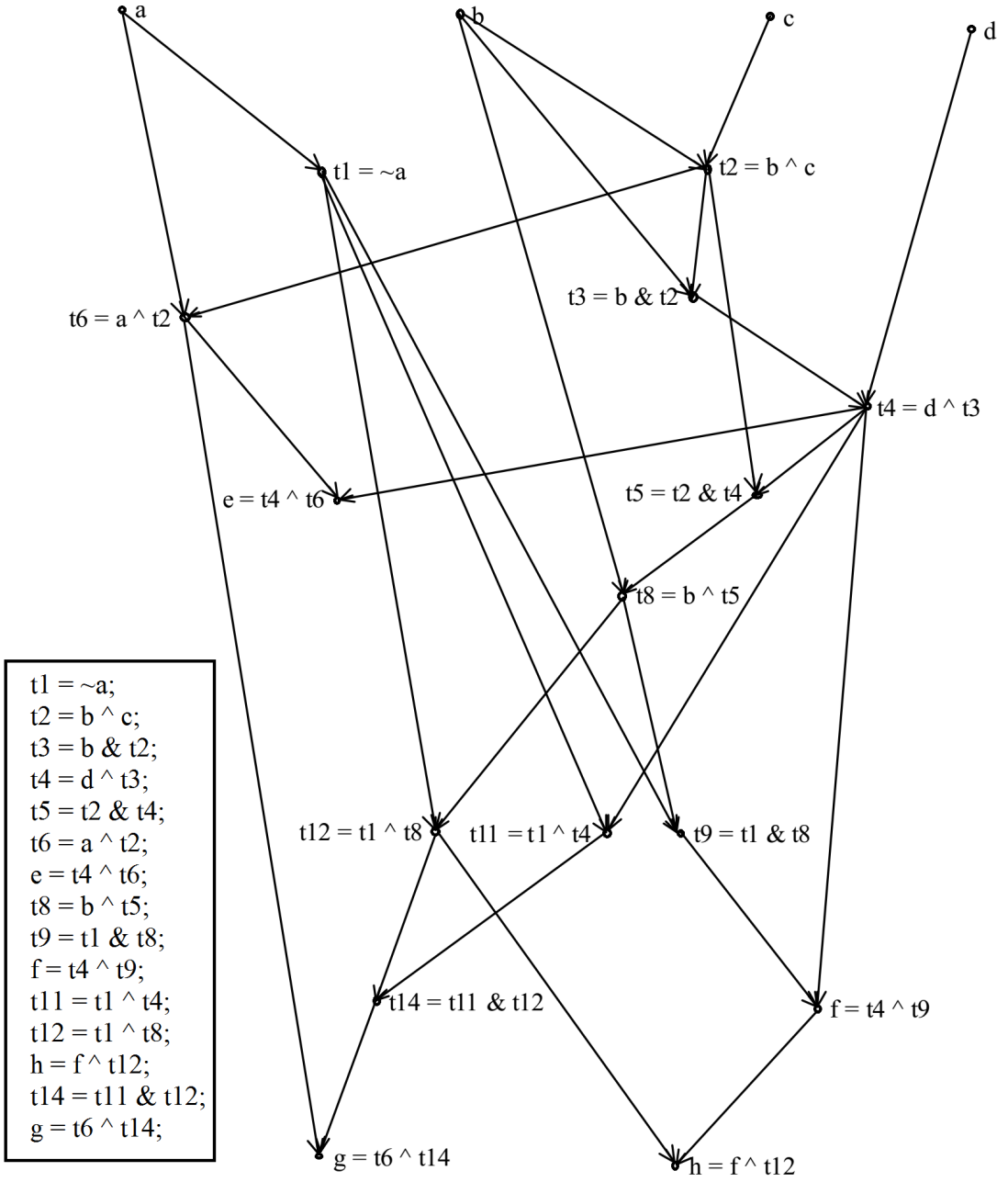

Figure 1: Diagram view of 15-term bit-sliced implementation of PRESENT&LED Sbox

```
// PRESENT & LED inverse Sbox
/*****************************************************************************************/
/*u4 S[16] = { 0x5, 0xe, 0xf, 0x8, 0xc, 0x1, 0x2, 0xd, 0xb, 0x4, 0x6, 0x3, 0x0, 0x7, 0x9, 0xa,  };
/*****************************************************************************************/
```

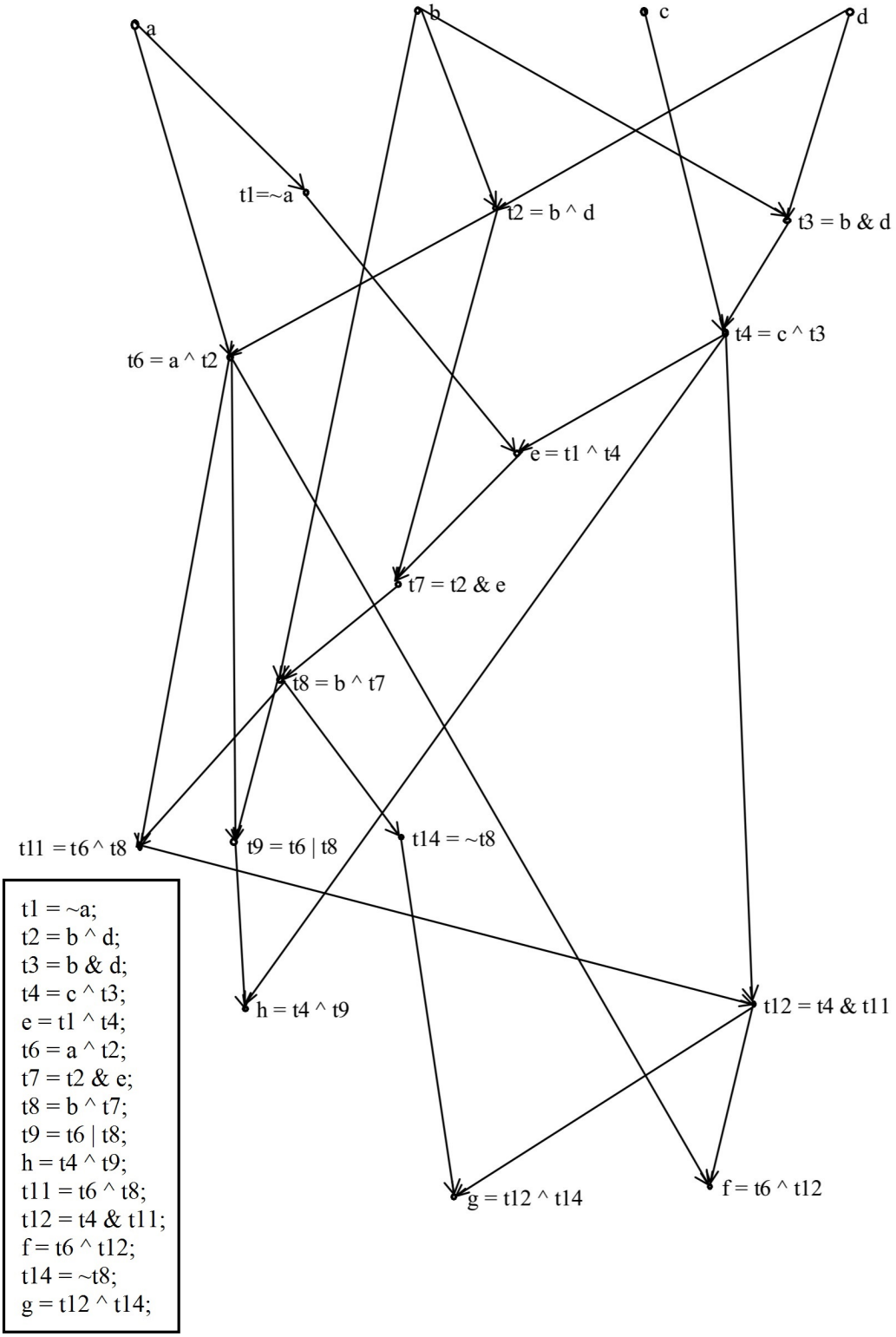

Figure 2: Diagram view of 15-term bit-sliced implementation of PRESENT&LED inverse Sbox

```
// PRINCE Sbox
/********************************************************************************/
/*u4 S[16] = { 0xb, 0xf, 0x3, 0x2, 0xa, 0xc, 0x9, 0x1, 0x6, 0x7, 0x8, 0x0, 0xe, 0x5, 0xd, 0x4,  };
/********************************************************************************/
```

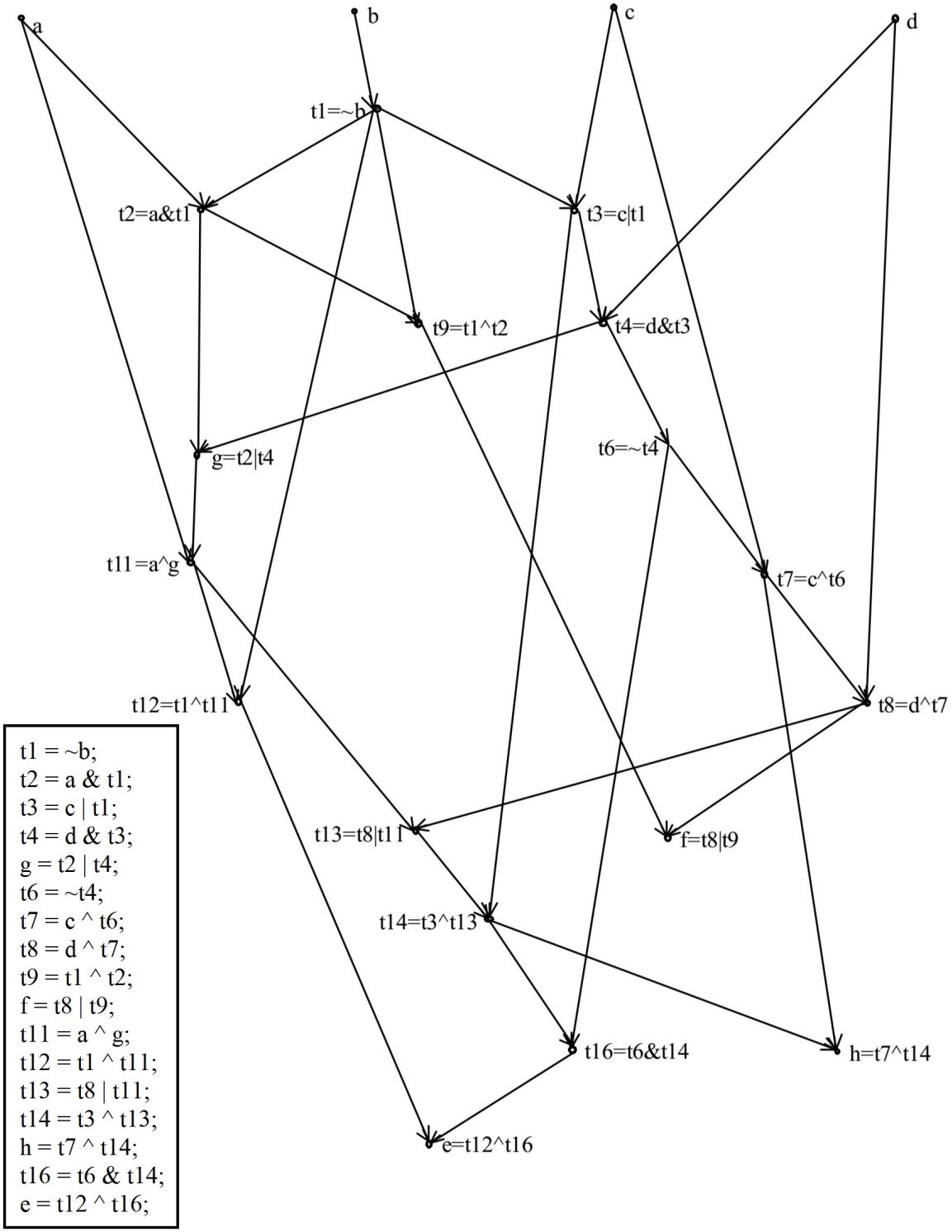

Figure 3: Diagram view of 17-term bit-sliced implementation of PRINCE Sbox

```
// PRINCE inverse Sbox
/*****************************************************************************/
/*u4 S[16] = { 0xb, 0x7, 0x3, 0x2, 0xf, 0xd, 0x8, 0x9, 0xa, 0x6, 0x4, 0x0, 0x5, 0xe, 0xc, 0x1,  };
/*****************************************************************************/
```

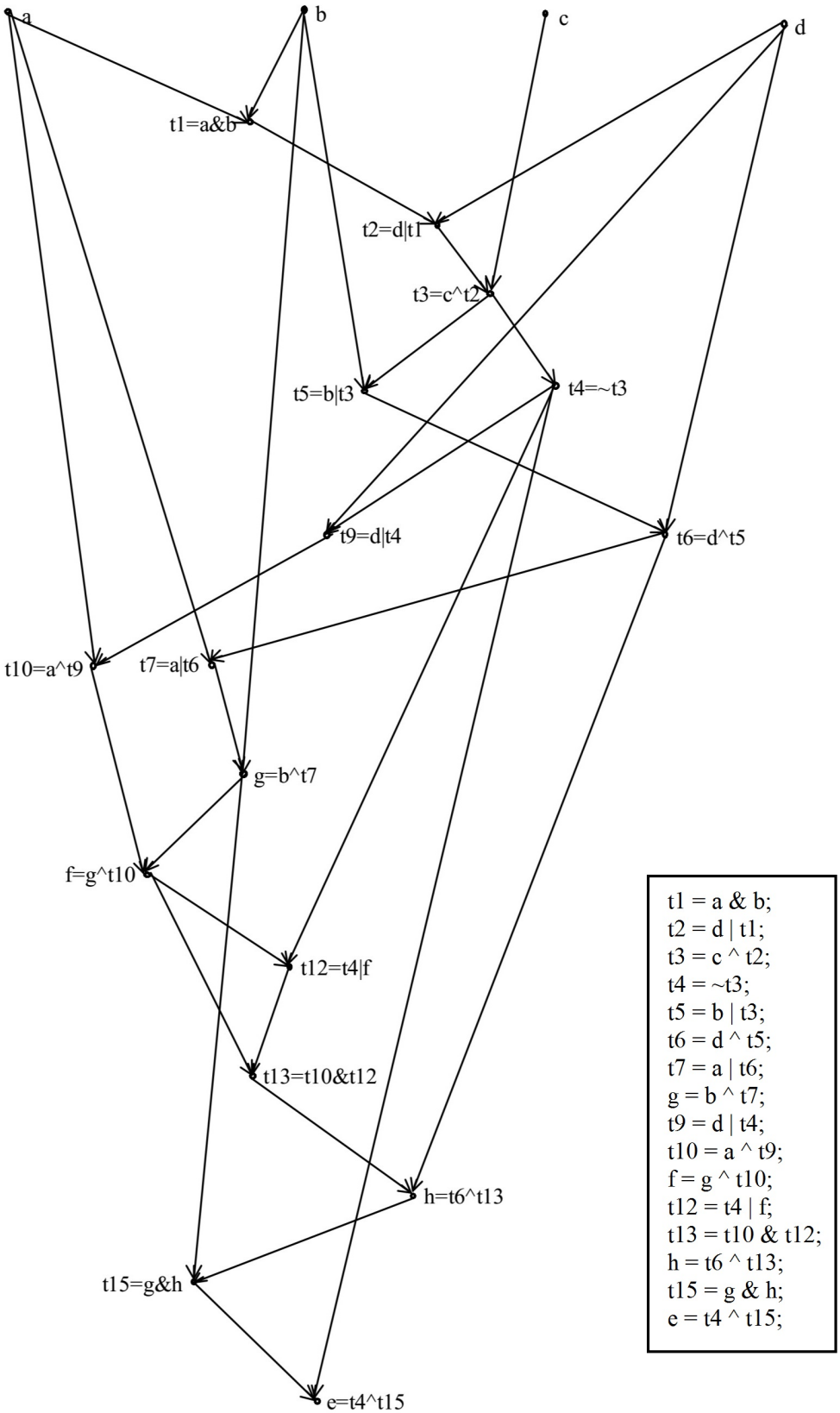

Figure 4: Diagram view of 16-term bit-sliced implementation of PRINCE inverse Sbox

```
// RECTANGLE Sbox
/*****************************************************************************/
/*u4 S[16] = { 0x6, 0x5, 0xc, 0xa, 0x1, 0xe, 0x7, 0x9, 0xb, 0x0, 0x3, 0xd, 0x8, 0xf, 0x4, 0x2,  };
/*****************************************************************************/
```

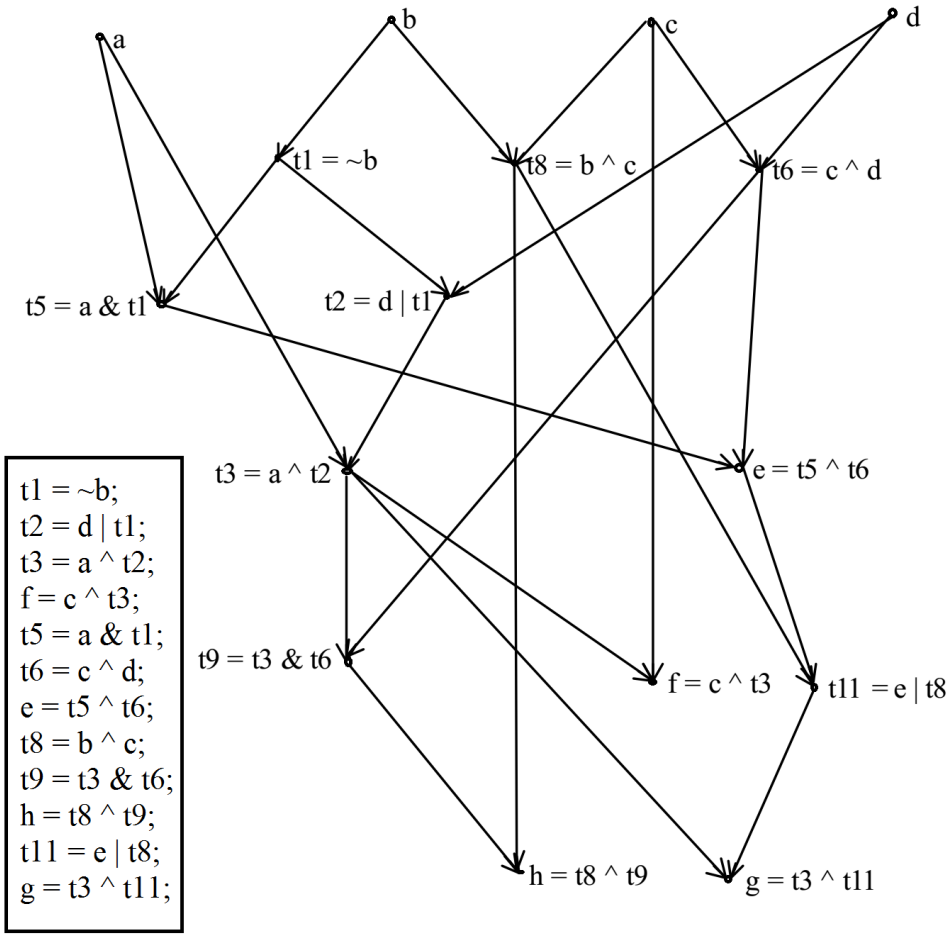

Figure 5: Diagram view of 12-term bit-sliced implementation of RECTANGLE Sbox

```
// RECTANGLE inverse Sbox
/*****************************************************************************/
/*u4 S[16] = { 0x9, 0x4, 0xf, 0xa, 0xe, 0x1, 0x0, 0x6, 0xc, 0x7, 0x3, 0x8, 0x2, 0xb, 0x5, 0xd,  };
/*****************************************************************************/
```

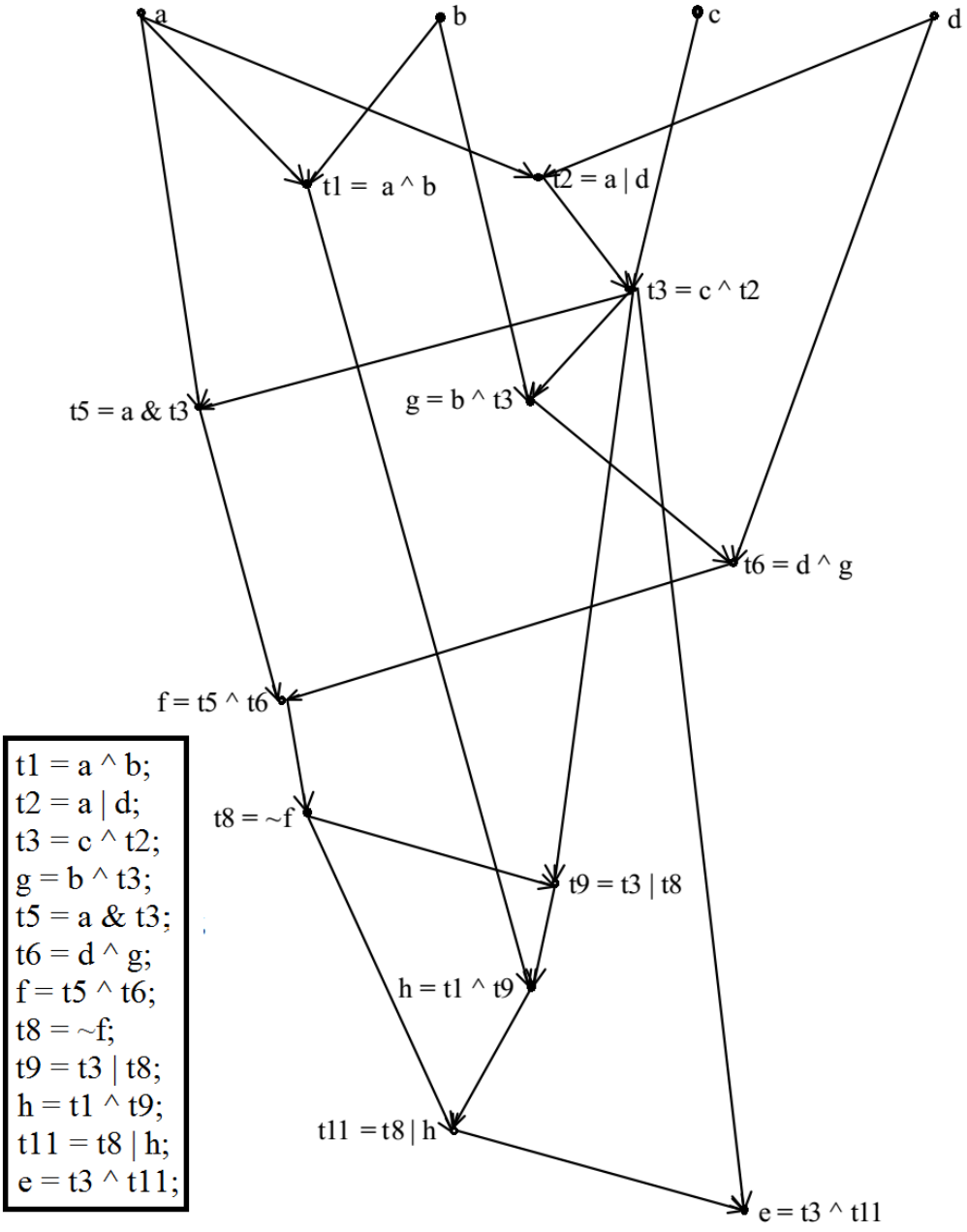

Figure 6: Diagram view of 12-term bit-sliced implementation of RECTANGLE inverse Sbox