1. $S_0 = 5$ a=14 b=15 m=21

$S_{0+1} = (14*5 +15) \bmod 21 > (70+15) \bmod 21 > 85 \bmod 21 >$ $S_1 = 1$

$S_{1+1} = (14*1 + 15) \bmod 21 > 29 \bmod 21 >$ $S_2 = 8$

$S_{2+1} = (14*8 + 15) \bmod 21 > (112+15) \bmod 21 > 127 \bmod 21 >$ $S_3 = 1$

$S_{3+1} = (14*1 + 15) \bmod 21 > 29 \bmod 21 >$ $S_4 = 8$

$S_{4+1} = (14*8 + 15) \bmod 21 > (112+15) \bmod 21 > 127 \bmod 21 >$ $S_5 = 1$

$S_{5+1} = (14*1 + 15) \bmod 21 > 29 \bmod 21 >$ $S_6 = 8$

$S_{6+1} = (14*8 + 15) \bmod 21 > (112+15) \bmod 21 > 127 \bmod 21 >$ $S_7 = 1$

$S_{7+1} = (14*1 + 15) \bmod 21 > 29 \bmod 21 >$ $S_8 = 8$

$S_{8+1} = (14*8 + 15) \bmod 21 > (112+15) \bmod 21 > 127 \bmod 21 >$ $S_9 = 1$

$S_{9+1} = (14*1 + 15) \bmod 21 > 29 \bmod 21 >$ $S_{10} = 8$

2.

CLK	FF4	FF3	FF2	FF1	FF0 - Output
0	0	0	1	1	1
1	0	0	0	1	1
2	1	0	0	0	1
3	0	1	0	0	0
4	1	0	1	0	0
5	1	1	0	1	0
6	0	1	1	0	1
7	0	0	1	1	0
8	0	0	0	1	1
9	1	0	0	0	1

10	**0**	**1**	**0**	**0**	**0**
11	**1**	**0**	**1**	**0**	**0**
12	**1**	**1**	**0**	**1**	**0**
13	**0**	**1**	**1**	**0**	**1**
14	**0**	**0**	**1**	**1**	**0**
15	**0**	**0**	**0**	**1**	**1**
16	**1**	**0**	**0**	**0**	**1**
17	**0**	**1**	**0**	**0**	**0**
18	**1**	**0**	**1**	**0**	**0**
19	**1**	**1**	**0**	**1**	**0**
20	**0**	**1**	**1**	**0**	**1**
21	**0**	**0**	**1**	**1**	**0**
22	**0**	**0**	**0**	**1**	**1**
23	**1**	**0**	**0**	**0**	**1**
24	**0**	**1**	**0**	**0**	**0**
25	**1**	**0**	**1**	**0**	**0**
26	**1**	**1**	**0**	**1**	**0**
27	**0**	**1**	**1**	**0**	**1**
28	**0**	**0**	**1**	**1**	**0**
29	**0**	**0**	**0**	**1**	**1**
30	**1**	**0**	**0**	**0**	**1**

Worked on next page. It is sort of….strange, but hopefully you understand it the way I understood it using that way.

CLK 0: FF2(1) and FF1(1) go into XOR to become 0. XOR of FF1/FF2 goes into XOR for FF3(0) and stays 0. FF4 for CLK1 becomes 0, all other numbers shift to right.

CLK1: FF2(0) XOR FF1(1) = 1	1 XOR FF3(0) = 1	CLK2 FF4 = 1
CLK2: FF2(0) XOR FF1(0) = 0	0 XOR FF3 (0) = 0	CLK3 FF4 = 0
CLK3: FF2(0) XOR FF1(0) = 0	0 XOR FF3 (1) = 1	CLK 4 FF4 = 1
CLK4: FF2(1) XOR FF1(0) = 1	1 XOR FF3 (0) = 1	CLK 5 FF4 = 1
CLK5: FF2(0) XOR FF1(1) = 1	1 XOR FF3 (1) = 0	CLK 6 FF4 = 0
CLK6: FF2(1) XOR FF1(0) = 1	1 XOR FF3 (1) = 0	CLK 7 FF4 = 0
CLK7: FF2(1) XOR FF1(1) = 0	0 XOR FF3 (0) = 0	CLK 8 FF4 = 0
CLK8: FF2(0) XOR FF1(1) = 1	1 XOR FF3 (0) = 1	CLK 9 FF4 = 1
CLK9: FF2(0) XOR FF1(0) = 0	0 XOR FF3 (0) = 0	CLK10 FF4 = 0
CLK10: FF2(0) XOR FF1(0) = 0	0 XOR FF3 (1) = 1	CLK 11 FF4 = 1
CLK11: FF2(1) XOR FF1(0) = 1	1 XOR FF3 (0) = 1	CLK 12 FF4 = 1
CLK12: FF2(0) XOR FF1(1) = 1	1 XOR FF3 (1) = 0	CLK 13 FF4 = 0
CLK13: FF2(1) XOR FF1(0) = 1	1 XOR FF3(1) = 0	CLK 14 FF4 = 0
CLK14: FF2(1) XOR FF1(1) = 0	0 XOR FF3(0) = 0	CLK 15 FF4 = 0
CLK15: FF2(0) XOR FF1(1) = 1	1 XOR FF3 (0) = 1	CLK 16 FF4 = 1
CLK16: FF2(0) XOR FF1(0) = 0	0 XOR FF3 (0) = 0	CLK 17 FF4 = 0
CLK17: FF2(0) XOR FF1(0) = 0	0 XOR FF3 (1) = 1	CLK 18 FF4 = 1
CLK18: FF2(1) XOR FF1(0) = 1	1 XOR FF3 (0) = 1	CLK 19 FF4 = 1
CLK19: FF2(0) XOR FF1(1) = 1	1 XOR FF3(1) = 0	CLK 20 FF4 = 0
CLK20: FF2(1) XOR FF1(0) = 1	1 XOR FF3(1) = 0	CLK 21 FF4 = 0
CLK21: FF2(1) XOR FF1(1) = 0	0 XOR FF3(0) = 0	CLK 22 FF4 = 0
CLK22: FF2(0) XOR FF1(1) = 1	1 XOR FF3(0) = 1	CLK 23 FF4 = 1
CLK23: FF2(0) XOR FF1(0) = 0	0 XOR FF3 (0) = 0	CLK 24 FF4 = 0
CLK24: FF2(0) XOR FF1(0) = 0	0 XOR FF3 (1) = 1	CLK 25 FF4 = 1
CLK25: FF2(1) XOR FF1(0) = 1	1 XOR FF3 (0) = 1	CLK 26 FF4 = 1
CLK26: FF2(0) XOR FF1(1) = 1	1 XOR FF3(1) = 0	CLK 27 FF4 = 0
CLK27: FF2(1) XOR FF1(0) = 1	1 XOR FF3(1) = 0	CLK 28 FF4 = 0
CLK28: FF2(1) XOR FF1(1) = 0	0 XOR FF3(0) = 0	CLK 29 FF4 = 0
CLK29: FF2(0) XOR FF1(1) = 1	1 XOR FF3(0) = 1	CLK 30 FF4 = 1

The cycle length is 7, with the actual output cycle being 1100010.

(Sorry for the lengthy shown work again)

3. ***Step 0: Convert D4C3C2A1 to Binary =***

D4C3B2A1 > 1101 0100 1100 0011 1011 0010 1010 0001

Step 1: Expansion takes the last bit of the former character, and first of next, and adds them ot the current bit

1101 0100 1100 0011 1011 0010 1010 0001 expanded becomes:

111010 101001 011000 000111 110110 100101 010100 000011

Step 2: Convert Key to Binary

F0D532A490C6 >

1111 0000 1101 0101 0011 0010 1010 0100 1001 0000 1100 0110

Step 3: Expanded 8-bit XOR Hexidecimal key and Split into S_{1-8}

1110 1010 1001 0110 0000 0111 1101 1010 0101 0101 0000 0011

XOR

1111 0000 1101 0101 0011 0010 1010 0100 1001 0000 1100 0110

0001 1010 0100 0011 0011 0101 0111 1110 1100 0101 1100 0101

Splits into:

S_1 – 000110 | S_2 – 100100 | S_3 – 001100 | S_4 – 110101 | S_5 – 011111 | S_6 – 101100 | S_7 – 010111

S_8 – 000101

Step 4: Convert to decimal using end-bits and center bits

S_1 – 000110 > End = 00 > Row 0 Center = 0011 > Col 3 > 1 becomes 0001

S_2 – 100100 > End = 10 > Row 2 Center = 0010 > Col 2 > 7 becomes 0111

S_3 – 001100 > End = 00 > Row 0 Center = 0110 > Col 6 15 becomes 1111

S_4 – 110101 > End 11 > Row 3 Center 1010 > Col 10 5 becomes 0101

S_5 – 011111 > End 01 > Row 1 Center 1111 > Col 15 6 becomes 0110

S_6 – 101100 > End 10 > Row 2 Center 0110 > Col 6 12 becomes 1100

S_7 – 010111 > End 01 > Row 1 Center 1011 > Col 11 12 becomes 1100

S_8 – 000101 > End 01 > Row 1 Center 0010 > Col 2 13 becomes 1101

The total binary output would be: 0001 0111 1111 0101 0110 1100 1100 1101.

Step 5: Permutation Matrix

0	0	0	1	0	1	1	1
1	1	1	1	0	1	0	1
0	1	1	0	1	1	0	0
1	1	0	0	1	1	0	1

(I tried to number it, but it broke the table 3 times. If it's ok with you, I did the permutation in my head)

Permutation Matrix:

1	1	0	1	1	1	0	0
0	0	0	1	0	1	0	1
0	1	0	1	1	0	0	1
1	0	1	1	1	1	1	1

The final 32-bit output is 1101 1100 0001 0101 0101 1001 1011 1111

After converting to Hex; it becomes DC1559BF.

Answer = DC1559BF