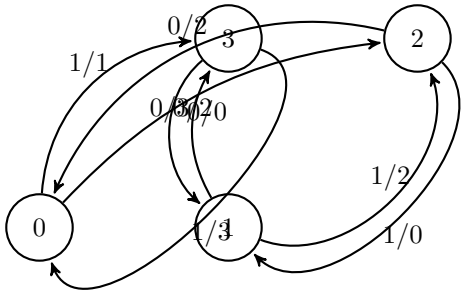

Zustand	Eingabe	Ausgabe	Folgezustand	Code Folgezustand			
				z3+	z2+	z1+	z0+
0	0	0	2	0	1	0	0
0	1	1	3	1	0	0	0
1	0	2	3	1	0	0	0
1	1	2	2	0	1	0	0
2	0	2	0	0	0	0	1
2	1	0	1	0	0	1	0
3	0	3	1	0	0	1	0
3	1	3	0	0	0	0	1

z3+

Zustand	Eingabe	Ausgabe	Folgezustand	Code Folgezustand			
				z3+	z2+	z1+	z0+
0	1	1	3	1	0	0	0
1	0	2	3	1	0	0	0

```
z3+ := (z0 and x) or (z1 and not x);
```

z2+

Zustand	Eingabe	Ausgabe	Folgezustand	Code Folgezustand			
				z3+	z2+	z1+	z0+
0	0	0	2	0	1	0	0
1	1	2	2	0	1	0	0

```
z2+ := (z0 and not x) or (z1 and x);
```

z1+

Zustand	Eingabe	Ausgabe	Folgezustand	Code Folgezustand			
				z3+	z2+	z1+	z0+
2	1	0	1	0	0	1	0

				Code Folgezustand			
				z3+	z2+	z1+	z0+
3	0	3	1	0	0	1	0

```
z1+ := (z2 and x) or (z3 and not x);
```

z0+

Zustand	Eingabe	Ausgabe	Folgezustand	Code Folgezustand			
				z3+	z2+	z1+	z0+
2	0	2	0	0	0	0	1
3	1	3	0	0	0	0	1

```
z0+ := (z2 and not x) or (z3 and x);
```

Zustand	Eingabe	Ausgabe	Folgezustand	Code Folgezustand			
				z3+	z2+	z1+	z0+
0	0	00	2	0	1	0	0
0	1	01	3	1	0	0	0
1	0	10	3	1	0	0	0
1	1	10	2	0	1	0	0
2	0	10	0	0	0	0	1
2	1	00	1	0	0	1	0
3	0	11	1	0	0	1	0
3	1	11	0	0	0	0	1

y1

Zustand	Eingabe	Ausgabe	Folgezustand	Code Folgezustand			
				z3+	z2+	z1+	z0+
1	0	10	3	1	0	0	0
1	1	10	2	0	1	0	0
2	0	10	0	0	0	0	1
3	0	11	1	0	0	1	0
3	1	11	0	0	0	0	1

```
y1 := (z1 and not x) or
        (not z1 and x) or
        (z2 and not x) or
        (z3 and not x) or
        (z3 and x);
```

y0

Zustand	Eingabe	Ausgabe	Folgezustand	Code Folgezustand			
				z3+	z2+	z1+	z0+
0	1	01	3	1	0	0	0
3	0	11	1	0	0	1	0
3	1	11	0	0	0	0	1

```
y0 := (z0 and x) or
        (z3 and not x0) or
        (z3 and x);


z3+ := (z0 and x) or (z1 and not x);
z2+ := (z0 and not x) or (z1 and x);
z1+ := (z2 and x) or (z3 and not x);
z0+ := (z2 and not x) or (z3 and x);


y1 := (z1 and not x) or
        (not z1 and x) or
        (z2 and not x) or
        (z3 and not x) or
        (z3 and x);

y0 := (z0 and x) or
        (z3 and not x0) or
        (z3 and x);



z3s <= (z0 and x) or (z1 and not x);
z2s <= (z0 and not x) or (z1 and x);
z1s <= (z2 and x) or (z3 and not x);
z0s <= (z2 and not x) or (z3 and x);
y1 <= (z1 and not x) or
        (not z1 and x) or
        (z2 and not x) or
        (z3 and not x) or
        (z3 and x);
y0 <= (z0 and x) or
        (z3 and not x0) or
        (z3 and x);


library ieee;
use ieee.std_logic_1164.all;

entity automat20240130 is
port (
    z3s, z2s, z1s, z0s: out std_logic;
    y1, y0: out std_logic;
    x: in std_logic;
```

```
    z3, z2, z1, z0: in std_logic
);
end;

architecture behaviour of automat20240130 is
begin
    z3s <= (z0 and x) or (z1 and not x);
    z2s <= (z0 and not x) or (z1 and x);
    z1s <= (z2 and x) or (z3 and not x);
    z0s <= (z2 and not x) or (z3 and x);
    y1 <= (z1 and not x) or
        (not z1 and x) or
        (z2 and not x) or
        (z3 and not x) or
        (z3 and x);
    y0 <= (z0 and x) or
        (z3 and not x) or
        (z3 and x);
end;

library ieee;
use ieee.std_logic_1164.all;

entity automat20240130testbench is
port (
    z3s, z2s, z1s, z0s: out std_logic;
    y1, y0: out std_logic
);
end;

architecture behaviour of automat20240130testbench is
    component automat20240130
    port (
        z3s, z2s, z1s, z0s: out std_logic;
        y1, y0: out std_logic;
        x: in std_logic;
        z3, z2, z1, z0: in std_logic
    );
    end component;
    signal x: std_logic;
    signal z3, z2, z1, z0: std_logic;
begin

    a: automat20240130 PORT MAP (z3s=>z3s, z2s=>z2s, z1s=>z1s, z0s=>z0s,
    y1=>y1, y0=>y0, x=>x, z3=>z3, z2=>z2, z1=>z1, z0=>z0);
    z0 <= '1' after 0 ns, '0' after 40 ns;
```

```
    z1 <= '0' after 0 ns, '1' after 40 ns, '0' after 80 ns;
    z2 <= '0' after 0 ns, '1' after 80 ns, '0' after 120 ns;
    z3 <= '0' after 0 ns, '1' after 120 ns, '0' after 160 ns;
    x <= '0' after 0 ns, '1' after 10 ns, '0' after 20 ns,
    '1' after 30 ns, '0' after 40 ns, '1' after 50 ns, '0' after 60 ns, '
    1' after 70 ns, '0' after 80 ns, '1' after 90 ns, '0' after 100 ns,
    '1' after 110 ns, '0' after 120 ns, '1' after 130 ns,
    '0' after 140 ns, '1' after 150 ns, '0' after 160 ns;


end;
```