- Übergangs-/Ausgangsschaltnetz

```
fsm: process (state, strt, len, i, i1, j, d, m, y, tmp, min, flg, swp, dib) is
     begin
       state0 <= state;
       i0     <= i;
       j0     <= j;
       m0     <= m;
       y0     <= y;
       tmp0   <= tmp;
       min0   <= min;
       d0     <= d;
       flg0   <= flg;
       swp0   <= swp;

       ofs    <= i;   -- default (OTHERS => '0');
       WEB    <= '0';
       ENB    <= '0';
       DOB    <= tmp; -- default (OTHERS => '0');

       case state is
         when S0 =>
           if strt='1' then
             d0     <= '0';
             i0     <= (others => '0');
             m0     <= len - 1;
             state0 <= S1;
           end if;
```

- Übergangs-/Ausgangsschaltnetz (Fortsetzung)

```
when S1 =>
   if i<m then
      -- ofs    <= i;
      ENB    <= '1';
      min0   <= i;
      flg0   <= '0';
      swp0   <= '0';
      j0     <= '0' & i1;
      state0 <= S2;
   else
      d0     <= '1';
      state0 <= S0;
   end if;
```

- Übergangs-/Ausgangsschaltnetz (Fortsetzung)

```
when S2 =>
  if flg='0' then
    flg0 <= '1';
    y0   <= DIB;
    tmp0 <= DIB;
  end if;
  if j<=m then
    ofs    <= j(ofs'range);
    ENB    <= '1';
    state0 <= S3;
  else
    if swp='1' then
      ofs    <= min;
      DOB    <= y;
      ENB    <= '1';
      WEB    <= '1';
      state0 <= S4;
    else
      i0     <= i1;
      state0 <= S1;
    end if;
  end if;
```

- Übergangs-/Ausgangsschaltnetz (Fortsetzung)

```
          when S3 =>
            if DIB<tmp then
              swp0   <= '1';
              min0   <= j(min0'range);
              tmp0   <= DIB;
            end if;
            j0     <= j + 1;
            state0 <= S2;

          when S4 =>
            -- ofs    <= i;
            -- DOB    <= tmp;
            ENB    <= '1';
            WEB    <= '1';
            i0     <= i1;
            state0 <= S1;
        end case;

    end process;

end verhalten;
```