

## Aufgabe 1

### Aufgabe 1

- (a) process (CLK, RST)

```
begin
    if (CLK'event and CLK= '1' and RST = '0' ) then
        Q <= 0;
    elsif (CLK'event and CLK='1') then
        Q <= D;
    end if;
end process;
```

- (b) Resource Sharing bezeichnet das Wiederverwenden von Komponenten, sofern diese nicht permanent beansprucht werden.

- (c) Ohne Resource Sharing werden 2 Addierer und 2 Multiplizierer benötigt.

```
signal S: BIT_VECTOR(1 downto 0) ;
signal A: SIGNED(31 downto 0) ;
signal B: SIGNED(31 downto 0) ;
signal C: SIGNED(31 downto 0) ;
signal D: SIGNED(31 downto 0) ;
signal Z: SIGNED(31 downto 0) ;
p0: process (S,A,B,C,D) begin
    case S(0) is
        when 0 =>
            //Z ergibt sich aus A und B
            X <= A;
            Y <= B;
        when 1 =>
            // Z Ergiebt sich aus C und D;
            X <= C;
            Y <= D;
    end case;

    case s(1) is
        when 0 =>
            //Addition der beiden Operanden
            Z <= X + Y;
        when 1 =>
            //Multiplikation der beiden Operanden
            Z <= X * Y;
    end case;
end process;
```

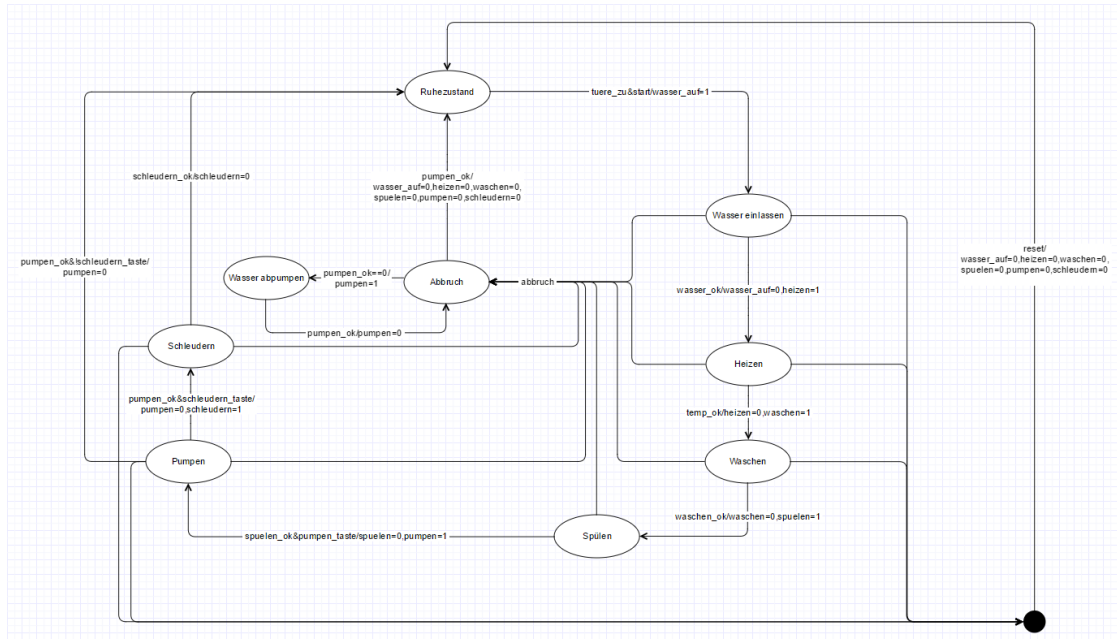
Wir haben hier den Multiplexer auseinandergezogen, um direkt auszuwählen, welche Operation mit welchen Operanden durchgeführt wird.

Durch Resource Sharing werden ein Addierer und ein Multiplizierer benötigt.

- (d) a) Explizites Steuerwerk mit implizitem Datenpfad  
b) Explizites Steuerwerk mit explizitem Datenpfad  
c) Implizites Steuerwerk mit implizitem Datenpfad

## Aufgabe 2

(b) FSM



(c) VCD Waves

