

Als Hilfsmittel sind erlaubt: - eigene Mitschriften, eigene Lösungen von Übungsaufgaben, ausgehändigte/empfohlene Skripte, Fachliteratur.

+++ **Es sind alle Bearbeitungsschritte darzustellen, sonst gibt es keine Punkte!** +++

Bitte versehen Sie diese sowie alle folgenden Seiten mit Namen und Matrikelnummer.

Bitte benutzen Sie keinen roten Stift!!!

Bearbeitungszeit: 90 min.

Aufgabe 1

30 Punkte

Eine synchrone sequentielle Schaltung bestehe aus 2 JK-Flip-Flops y_1, y_0 , einem Eingang x und einem Ausgang z . Die Beschaltung der Flip-Flops ist (' = NOT, * = AND, + = OR):

$$J_1 = x' + y_0$$

$$K_1 = x' \cdot y_0'$$

$$J_0 = x' + y_1'$$

$$K_0 = x' \cdot y_1$$

Die Beschaltung des Ausgangs ist:

$$z = (x \cdot y_0) + (x' \cdot y_0')$$

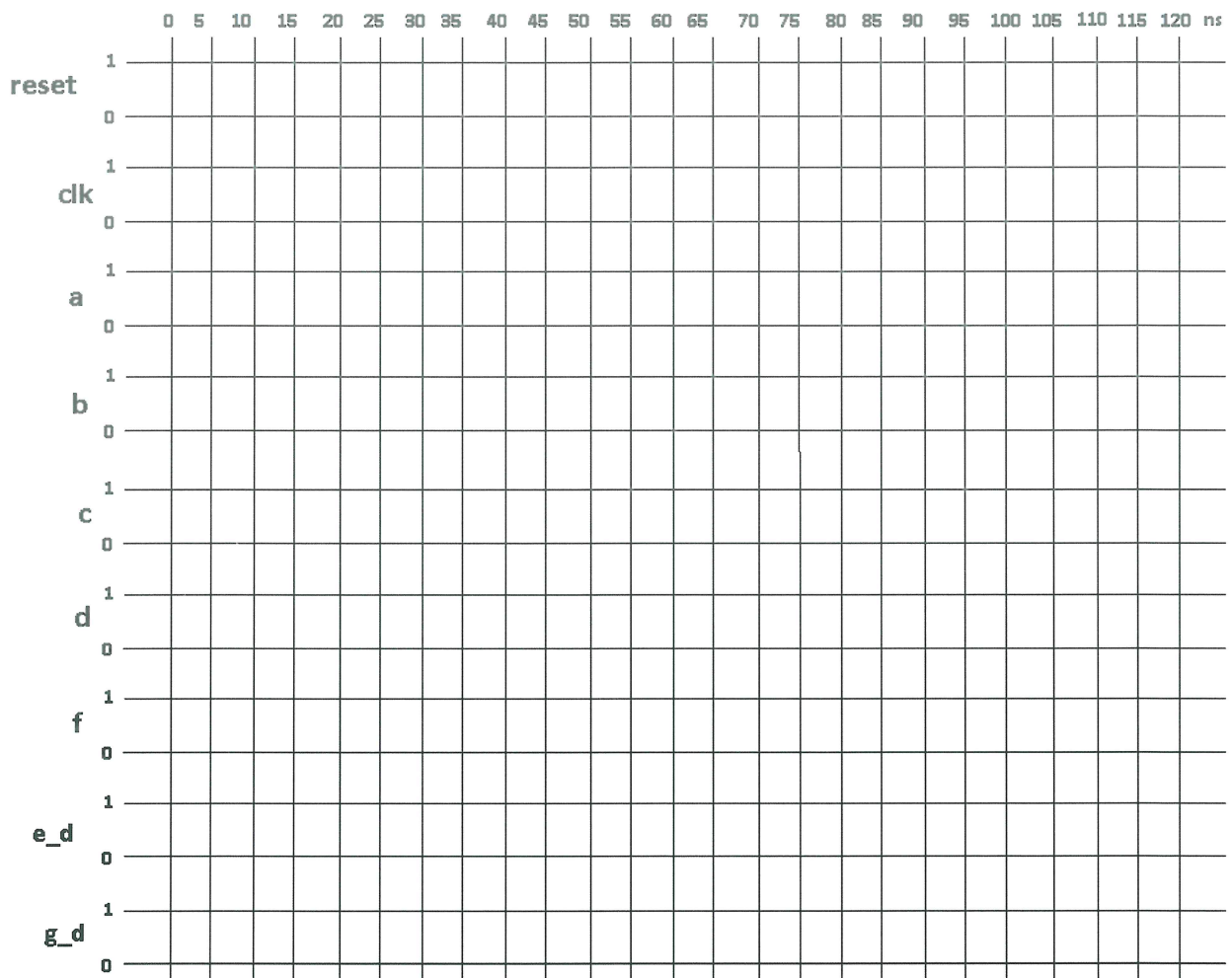
1.1 Leiten Sie den Zustandsgraphen dazu ab.

Bitte beachten: A = 00 B = 11 C = 10 D = 01

1.2 Realisieren Sie Schaltung mit T- Flip-Flops (mit Skizze!!!)

1.3 Beschreiben Sie die Schaltung in VHDL (Zwei Prozess Lösung!)

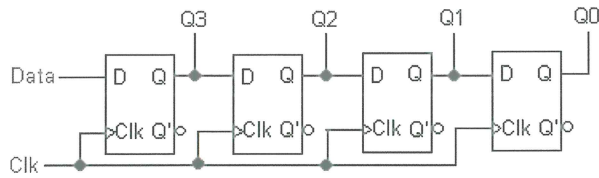
Füllen Sie hierfür die zeitlichen Verläufe der Signale (Variablen) clk, reset, a, b, c, d, e_d, g_d und f ein:



Aufgabe 2

10 Punkte

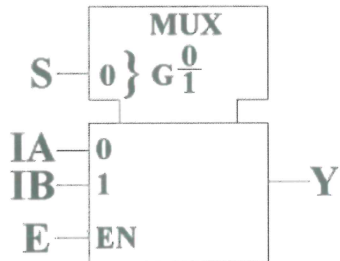
Gegeben sei folgende Ihnen bereits bekannte Schaltung:



Beschreiben Sie die Schaltung in VHDL (Verhaltensbeschreibung) unter Verwendung der FOR- Schleife

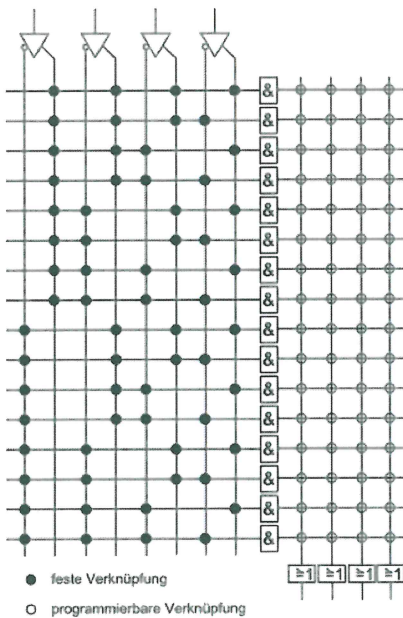
Aufgabe 3

10 Punkte



Realisieren Sie die Funktionen des vorgegeben Multiplexeres in dem unten stehenden PROM durch Ankreuzen der zu programmierenden Verknüpfungen. Schreiben Sie die programmierten Verknüpfungen ebenfalls als Gleichung auf.

Es sind alle Arbeits- und Ableitungsschritte zu dokumentieren.



Aufgabe 4

20 Punkte

Der folgende VHDL-Quelltext sei gegeben:

```
.....  
LIBRARY ieee;  
USE ieee.std_logic_1164.all;  
USE ieee.std_logic_arith.all;  
  
ENTITY Klausur IS  
  
END ENTITY Klausur;  
  
--  
ARCHITECTURE arch OF Klausur IS  
SIGNAL a, b: std_logic:= '1';  
SIGNAL d, f, g_d, e_d: std_logic:= '0';  
SIGNAL reset: std_logic := '1';  
SIGNAL clk: std_logic := '1';  
  
BEGIN  
a <= '0' after 2 ns, '1' after 11 ns, '0' after 29 ns, '1' after 48 ns, '0' after 56 ns, '1' after 75 ns, '0' after 102 ns;  
reset <= '0' after 12 ns, '1' after 19 ns, '0' after 105 ns;  
clk <= not clk after 10 ns;  
  
e_d <= a after 14 ns;  
g_d <= TRANSPORT a after 14 ns;  
  
PROCESS(clk, reset)  
variable c: std_logic:= '0';  
BEGIN  
IF (reset = '0') THEN  
b <= '0';  
d <= '0';  
c:= '1';  
ELSIF (clk = '1' and clk'event) THEN  
b <= a;  
d <= a;  
c := not b;  
d <= c;  
END IF;  
END PROCESS;  
  
PROCESS(d)  
BEGIN  
f <= a;  
END PROCESS;  
  
END ARCHITECTURE arch;  
*****
```