

Technische Grundlagen der Informatik

Vorrechenübung 13.12.12



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Was gibt es heute?

- Klausurregeln
- Vorrechenübung, Aufgaben aus alten Klausuren
- Nicht nur stumpfes Vorrechnen, aktive Teilnahme, stellen Sie Fragen!
- Nächste Woche keine Übungen

Klausurregeln

- 18.12.11, 18:00
- Dauer: 90min
- Raumverteilung steht am Montag im Moodle
- Bitte pünktlich sein!
- Wir werden die Klausur zuerst gemeinsam durchgehen, dann erst startet die Bearbeitungszeit

Klausurregeln - Fachliches

- Relevanter Stoff: Vorlesungen bis Kapitel 5, Übungen bis 8
- KEINE Hilfsmittel (Ausnahme: Wörterbuch für ausländische Studierende) – außer Stift und Lineal
- Verilog-Syntaxblatt wird der Klausur beiliegen
- Gattersymbole: Entweder DIN oder amerikanisch, aber nicht gemischt
- K-Diagramme: Wenn vorgegebene Belegung, muss das vorgegebene Diagramm benutzt werden
- Bewertet wird nicht nur die Lösung, auch der Lösungsweg

Klausurregeln - Formales

- Dokumentenechter Stift, blau oder schwarz, KEIN rot oder grün, auf KEINEN FALL Bleistift (auch nicht zum Vorzeichnen!!!!)
- Studentenausweis und Lichtbildausweis
- Kein eigenes Papier, wer mehr benötigt bekommt von uns, darauf Namen und Matrikelnummer & vorne bei der Aufgabe vermerken
- Durchgestrichenes DÜRFEN wir nicht bewerten – auch wenn es korrekt ist
- Essen und Trinken erlaubt, aber bitte Rücksicht (keine Popcorntüte 😊)

Klausurregeln - Formales

- BetrugsVERSUCH führt ohne Vorwarnung zum Ausschluss von der Prüfung
- In schweren Fällen Sanktionen bis hin zur Exmatrikulation

Klausurregeln – Bewertung

- Bsc INF PO 2009
- In der ersten Teilklausur 90 Punkte
- In der zweiten Teilklausur 90 Punkte
- Im Praktikum 45 Punkte
- Punkte werden einfach addiert (225 möglich)
- Mit 112,5 Punkten (50%) auf jeden Fall bestanden

Ergebnisse

- Die Ergebnisse der Korrektur werden im Moodle bekannt gegeben
- Sie werden veröffentlicht, wenn die Korrektur fertig ist, Nachfragen per Mail/PM/sonstwie beschleunigen es nicht!
- Einsicht nach der zweiten Teilklausur für beide Teile zusammen

Beispielaufgaben

- Die folgenden Aufgaben sind aus alten Klausuren und sollen typische Probleme / Fehler verhindern
- Deckt nicht den kompletten Stoff der Klausur ab
- Was heute gezeigt wird kommt nicht unbedingt in der Klausur dran (und was nicht gezeigt wird kann trotzdem vorkommen)
- Bei Problemen/Fragen: Nutzen Sie die Sprechstunden!

Aufgabe 6: Verilog (15 Punkte)

- Beschreiben Sie ein Schieberegister in Verilog. Es ist die folgende Spezifikation gegeben:
- a) Das Register soll 16 Bit breit sein
- b) Nach außen sollen die folgenden Ein- und Ausgänge zur Verfügung stehen:
 - clk: Das Taktsignal
 - reset: wenn reset=1, soll das Register auf 0 gesetzt werden. Das Signal soll synchron sein.
 - dataout[15:0]: Datenausgang
 - datain[15:0]: Dateneingang
 - write: wenn write=1, soll der Dateneingang in das Register übernommen werden
 - shiftright: wenn shiftright=1, soll der Inhalt des Registers arithmetisch um 1 nach rechts geschoben werden
 - shiftright: wenn shiftright=1, soll der Inhalt des Registers arithmetisch um 1 nach rechts geschoben werden
 - enable: siehe unten
- c) Sie können davon ausgehen, dass write, shiftright und shiftright nie gleichzeitig gesetzt sind.
- d) Am Datenausgang soll nur dann der Inhalt des Registers anliegen, wenn enable=1. Ansonsten soll 0 anliegen.
- Kommentieren Sie Ihren Code!

Aufgabe 6: Verilog Lösung

```
module shiftreg (  
  
    input wire      clk,  
    input wire      reset,  
    output wire [15:0] dataout,  
    input wire [15:0] datain,  
    input wire      write,  
    input wire      shiftright,  
    input wire      shiftright,  
    input wire      enable  
);  
  
    reg [15:0] shifter;           //Speicherelemente  
  
    always@(posedge clk) begin  
        if ( reset )  
            shifter <= 16 'b0;           //Rest, alles auf 0  
        else begin  
            if ( write )  
                shifter <= datain;       //Dateneingang übernehmen  
            if ( shiftright )  
                shifter <= { shifter [14:0] ,1'b0 }; //Linksshift  
            if ( shiftright )  
                shifter <= { shifter [15] , shifter [15:1] }; //arithmetischer Rechsshift  
        end  
    end  
  
    assign dataout = enable ? shifter : 16 'b0 ; // Ausgang nur bei enable gueltig  
endmodule
```

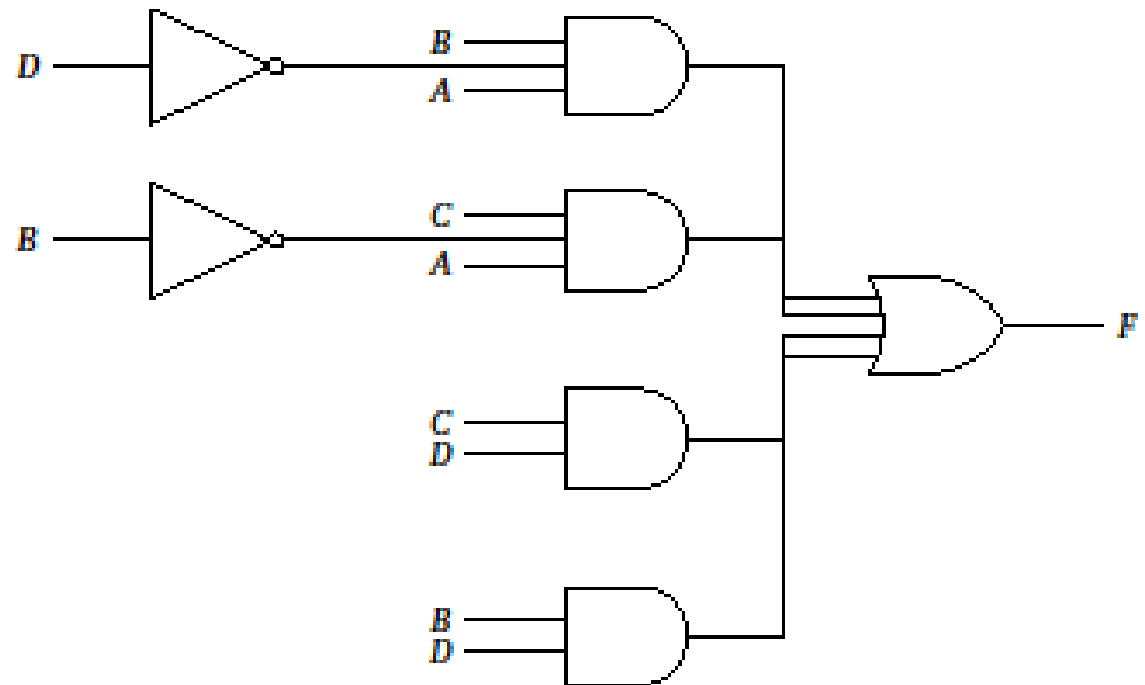
Aufgabe 4: Glitches (14 Punkte)

- Gegeben ist die folgende Gleichung:

$$F = \overline{D}BA + DC + DB + C\overline{B}A$$

- Zeichnen Sie einen Schaltplan für die Schaltung, die diese Funktion implementiert. Als Eingänge stehen nur die Variablen zur Verfügung, nicht ihre Komplemente. Es stehen Ihnen beliebige Gatter zur Verfügung.
- Ermitteln Sie mit Hilfe eines K-Digramms, ob die Schaltung potentiell Glitches enthält.
- Simulieren Sie den Übergang $ABC\overline{D} \rightarrow \overline{A}BC\overline{D}$, jedes Gatter hat eine Ausbreitungsverzögerung von 5ns.
- Geben Sie eine glitchfreie Funktion an.

Aufgabe 4 - Schaltplan



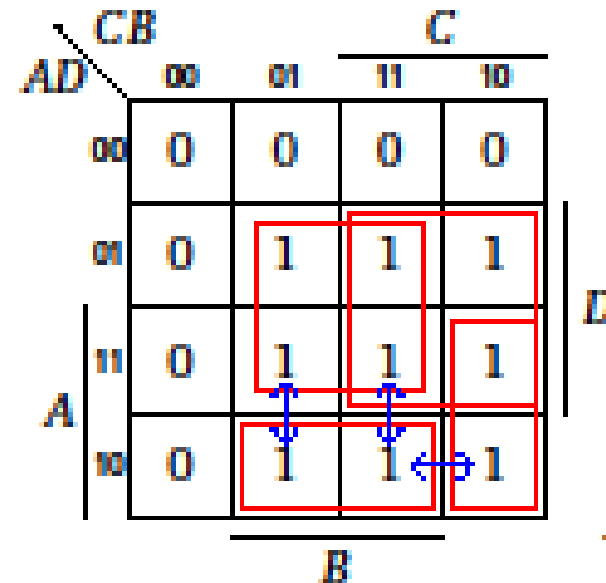
$$F = \overline{D}BA + DC + DB + C\overline{B}A$$

Aufgabe 4 - Glitches

$$F = \overline{D}BA + DC + DB + C\overline{B}A$$

- An den blau markierten Übergängen können Glitches auftreten
- Eine glitchfreie Funktion ist

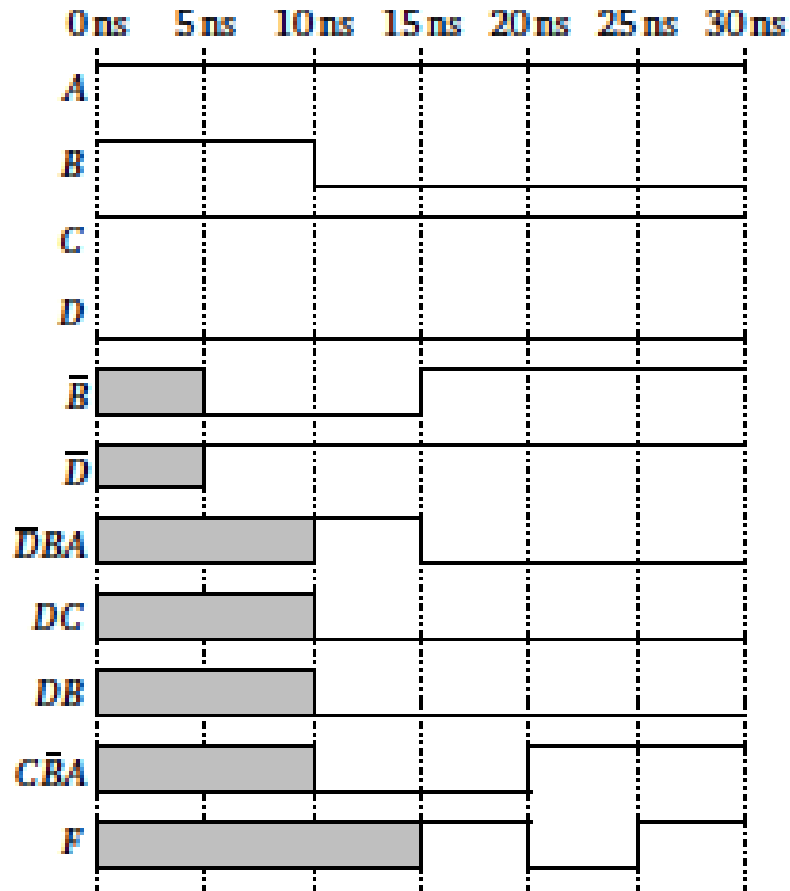
$$F = BA + DC + DB + CA$$



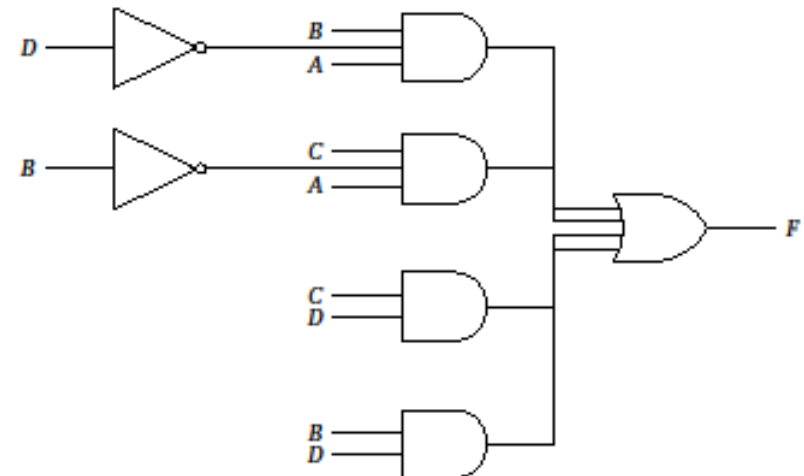
The Karnaugh map shows the function F with variables A, B, C, and D. The map is a 4x4 grid with rows labeled AD (00, 01, 11, 10) and columns labeled CB (00, 01, 11, 10). The function values are 0 for (AD, CB) pairs (00,00), (00,01), (00,11), (00,10), (01,00), (11,00), and (10,00). The function values are 1 for (AD, CB) pairs (01,01), (01,11), (01,10), (11,01), (11,11), (11,10), (10,01), (10,11), and (10,10). Red boxes group the 1s into four prime implicants: a 2x2 square (01,01)-(01,11)-(11,01)-(11,11), a 2x2 square (01,11)-(01,10)-(11,11)-(11,10), a 2x2 square (10,01)-(10,11)-(10,10)-(11,10), and a 2x2 square (10,01)-(10,11)-(10,10)-(10,10). Blue arrows indicate transitions between adjacent 1s: a vertical arrow from (10,01) to (11,01), a vertical arrow from (11,01) to (11,11), a vertical arrow from (11,11) to (11,10), a horizontal arrow from (10,01) to (10,11), a horizontal arrow from (10,11) to (10,10), and a horizontal arrow from (10,11) to (11,11).

AD \ CB	CB		C	
	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	0	1	1	1
10	0	1	1	1

Aufgabe 4: Timing-Diagramm



$$ABCD \rightarrow \bar{A}\bar{B}\bar{C}\bar{D}$$



Aufgabe 5: Automat

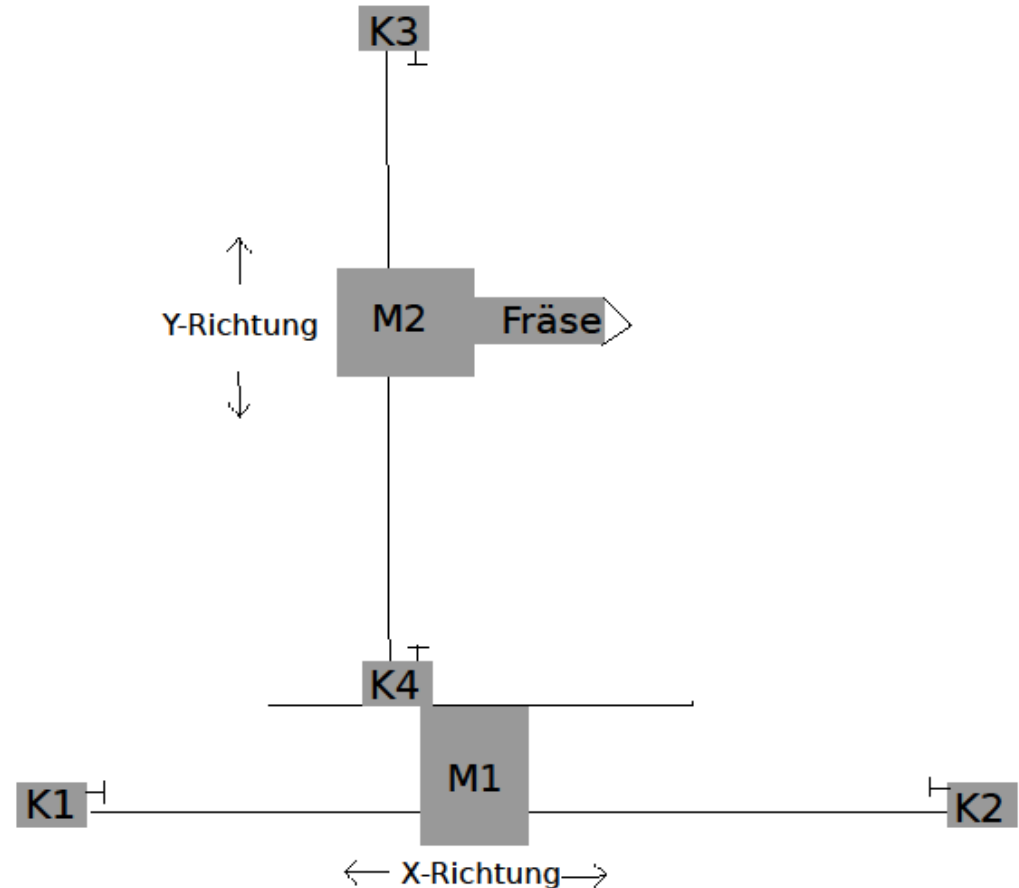
- Entwerfen Sie einen Mealy-Automaten zur Steuerung einer Fräse. Die folgende Spezifikation ist gegeben:
- Die Fräse hat 2 Motoren, einen für die Bewegung des Fräsenkopfes in X-Richtung ($M1$) und einen für die Bewegung in Y-Richtung ($M2$). Jeder Motor hat 2 Anschlüsse: einen für Vorwärtslauf ($M1v$) und einen für Rückwärtslauf ($M1r$).
- Für die Fertigung eines Werkstückes ist immer der folgende Ablauf nötig: Nach dem Einspannen des Holzes wird durch Drücken des Schalters $S1$ die Maschine aktiviert. Zuerst muss die Fräse in X-Richtung fräsen, bis der Anschlag erreicht ist (signalisiert durch den Kontaktschalter $K2$). Danach wird in Y-Richtung gearbeitet, auch hier wird das Ende durch einen Kontaktschalter $K3$ signalisiert. Nach Abschluss des Fräsvorgangs muss die Maschine wieder in die Startposition zurückkehren (angezeigt über die Kontaktschalter $K1$ und $K4$), danach wird das Ende durch eine Signallampe (L) angezeigt.
- Sie können davon ausgehen, dass die Maschine immer korrekt arbeitet und die Kontakte nur erwartete Werte liefern. Es darf immer nur ein Motor aktiviert sein.

Aufgabe 5: Automat

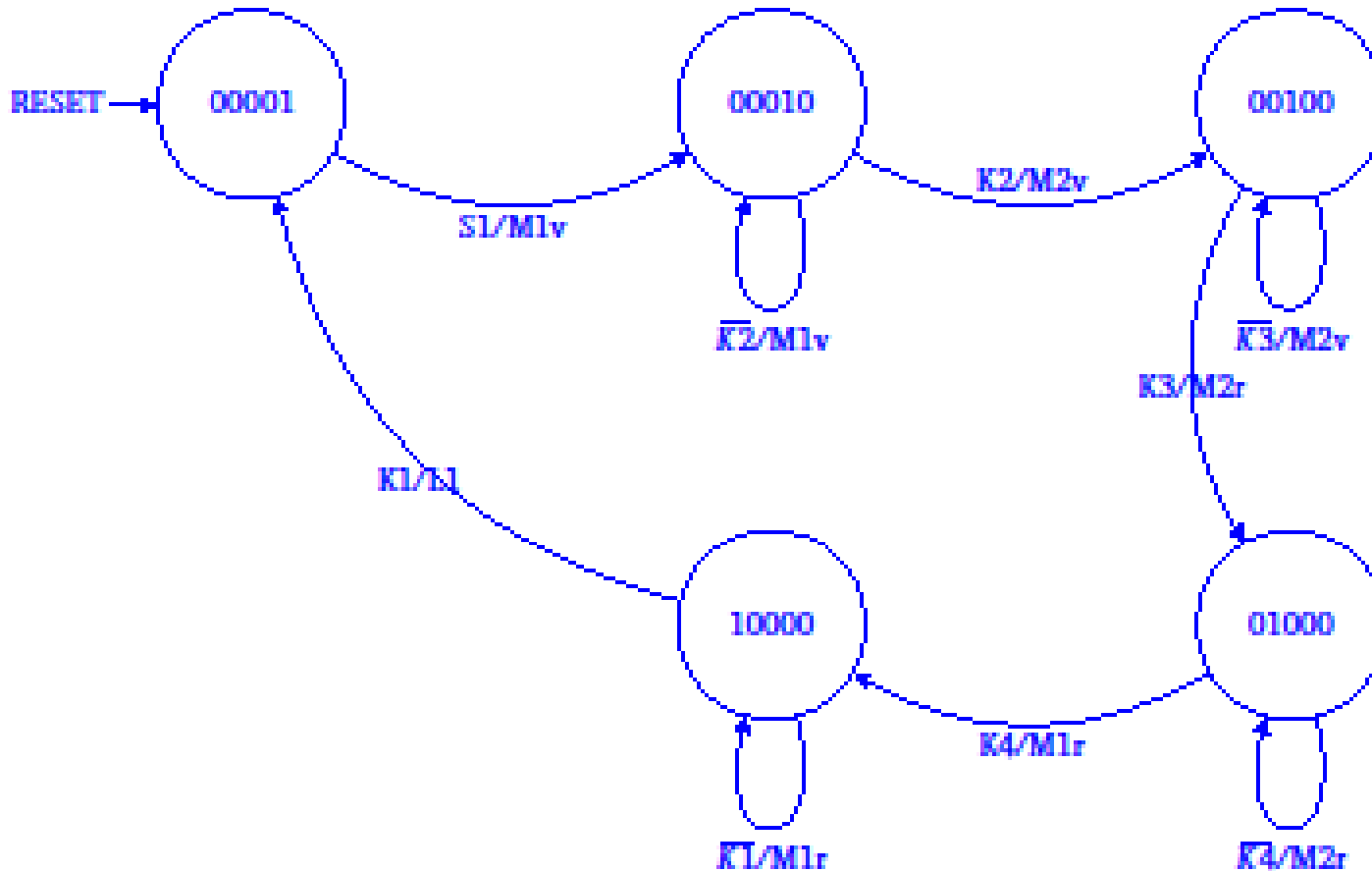
a) Geben Sie das Zustandsübergangdiagramm des Mealy-Automaten an. Verwenden Sie möglichst wenig Zustände.

b) Geben Sie die Zustandsübergangstabelle und die Ausgangstabelle an. Verwenden Sie One-Hot-Codierung für die Zustände (1-aus-N-Code). Hinweis: Nutzen Sie Don't Cares aus!

c) Geben Sie die Zustandsübergangsgleichungen und die Ausgangsgleichungen an.



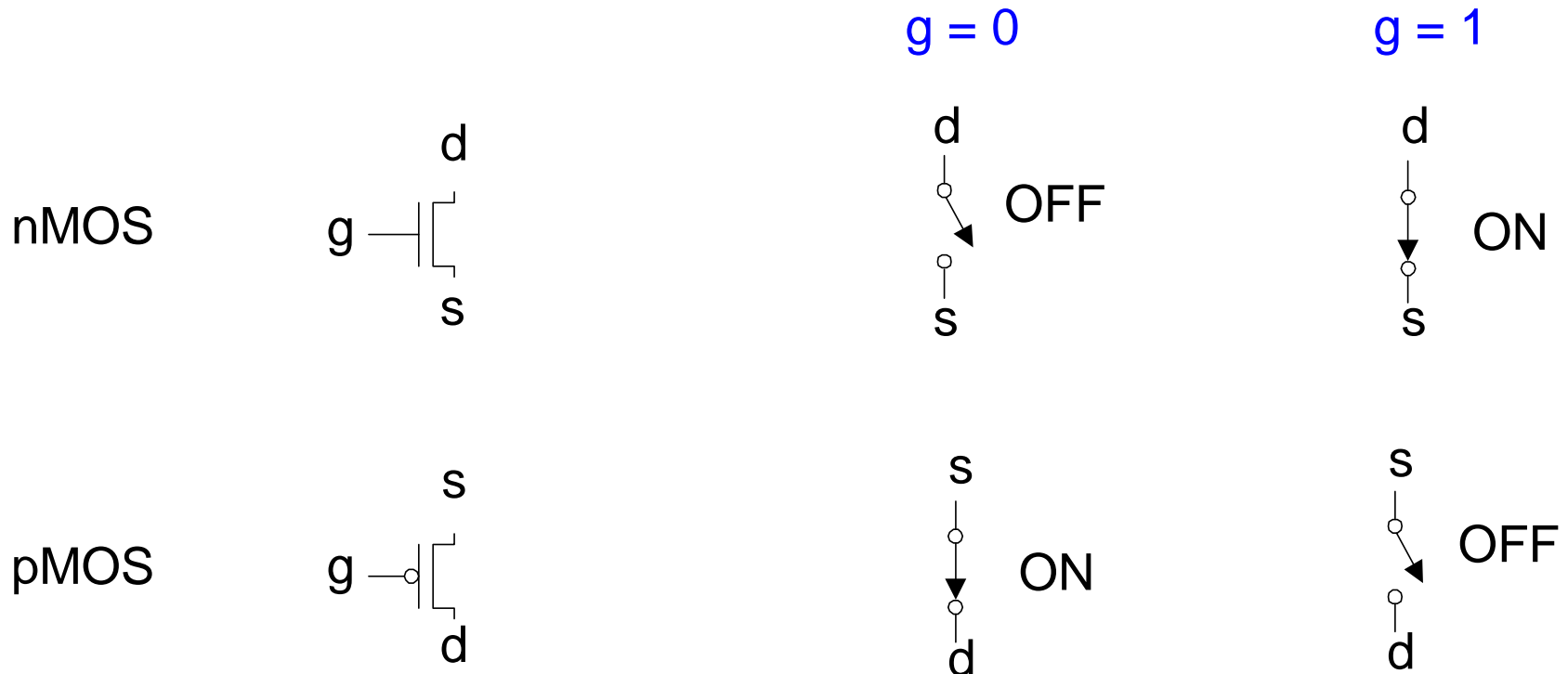
Aufgabe 5: Zustandsübergangsdigramm



Aufgabe 5: Zustandsübergangstabelle

- Kommt noch als extra-pdf ins Moodle

Aufgabe 3: nMOS und pMOS



nMOS Transistoren leiten 0'en **gut** zwischen S und D weiter
pMOS Transistoren leiten 1'en **gut** zwischen S und D weiter

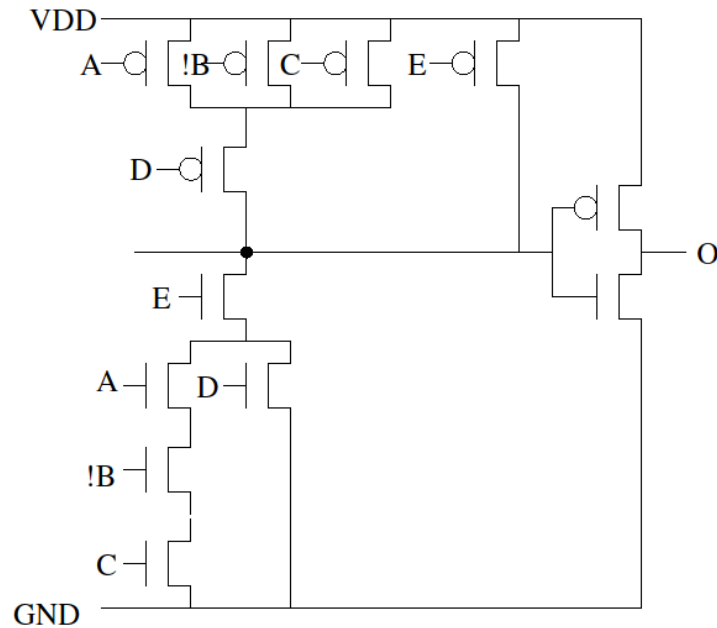
Aufgabe 3: nMOS und pMOS

- Realisieren Sie die folgende Funktion nur mit nMOS und pMOS Transistoren
- Es stehen Ihnen als Eingänge die Variablen, ihre Komplemente sowie logisch 0 und 1 zur Verfügung.

$$O = (\overline{A}BC + D)E$$

Aufgabe 3: Lösung

$$O = (\overline{A}BC + D)E$$



Noch Fragen?

- Jetzt stellen

- Oder die Sprechstundentermine vor der Klausur nutzen (ruhigere Atmosphäre, mehr Zeit für individuelle Betreuung):
 - Fr 14.12. 9:30 - 10:30 E102
 - Mo 17.12. 9:00 - 12:00 A126
 - Di 18.12. 10:00 - 11:00 E202

- Moodle-Forum nutzen

- Viel Erfolg bei der Prüfung