

Klausur

Einführung in die Rechnerarchitektur

Prof. Dr. Arndt Bode

Wintersemester 2012/2013

13. Februar 2013

Name: _____

Vorname: _____

Matrikelnummer: _____

Geburtsdatum: _____

Hörsaal: _____

Platz: _____

Unterschrift: _____

Ergebnis:

| Aufgabe | 1 | 2 | 3 | 4 | Ges. | Note |
|-----------|---|---|---|---|------|------|
| Punkte | | | | | | |
| Korrektur | | | | | | |

Hinweise zu den Aufgaben:

- Die Bearbeitungszeit beträgt 120 Minuten.
- Es sind keinerlei Hilfsmittel zugelassen, auch keine Taschenrechner.
- Versehen Sie dieses Angabenblatt auf der Titelseite mit Ihrem Namen, Vornamen und Matrikelnummer.
- Diese Angabe umfasst 24 bedruckte Seiten (inklusive Deckblatt).
- Außerdem erhalten Sie die folgende Merkblätter:

Anlage I: "Die wichtigsten 80386 Befehle und ihre Operanden"

Anlage II: "Mikroprogrammierung"

Anlage III: "Die wichtigsten VHDL Konstrukte"

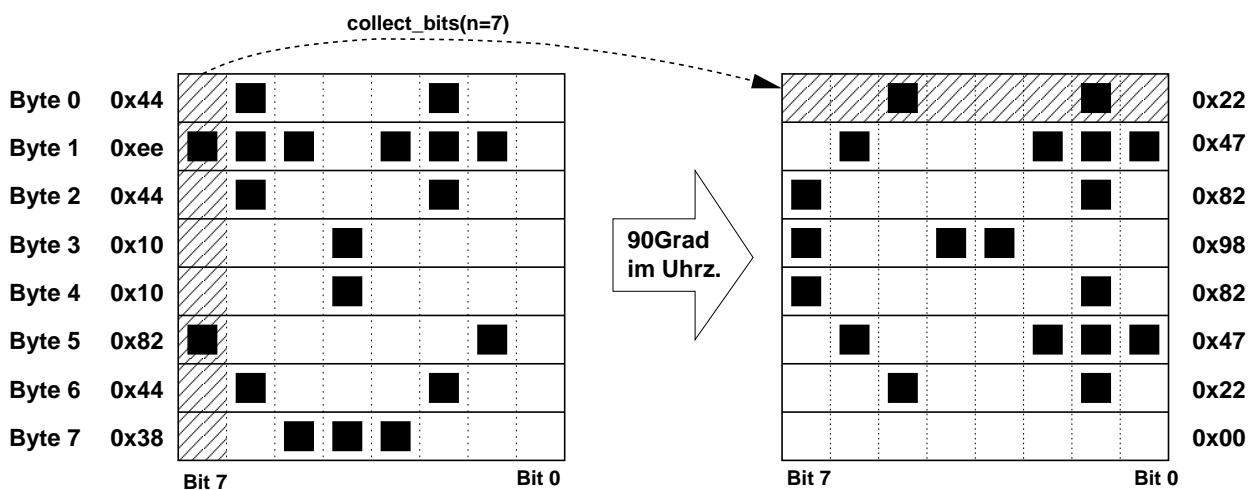
- Alle Lösungen sind in dieses Heft einzutragen. Sollte der vorgesehene Platz nicht ausreichen, so finden Sie am Ende weitere Blätter. Sollten auch diese nicht ausreichen, so wenden Sie sich bitte an die Aufsichten.
- Notizpapier wird auf Ihre Anfrage ausgegeben. Die Verwendung von eigenem Papier ist nicht gestattet.

Lösungen auf Notizpapier werden NICHT gewertet!

2.2 Bitmuster-Rotation

In dieser Aufgabe soll schrittweise ein Programm entwickelt werden, das den Inhalt von 8 aufeinanderfolgenden Bytes so verändert, dass die 8*8 Bitmatrix um 90 Grad im Uhrzeigersinn gedreht wird.

Beispiel (ein weißes Quadrat entspricht einem 0-Bit, ein schwarzes einem 1-Bit):



Es soll folgende Herangehensweise benutzt werden:

- Ein Unterprogramm `collect_bits` mit dem Parameter `n` "sammelt" über alle 8 Bytes die Bits der Spaltenposition `n` ein und gibt sie in einem Register zurück. Ein Beispiel für `n=7` ist in obiger Grafik grau schraffiert.

- Ein Unterprogramm rotate ruft collect_bits mit den Werten 7 bis 0 auf und speichert die Rückgabewerte entsprechend zurück.

Die folgenden Teilaufgaben sind unabhängig voneinander lösbar, wenn die jeweiligen Aufrufkonventionen beachtet werden.

2.2.1 Gesamtrotation

Schreiben Sie zunächst das 80386-Unterprogramm rotate, das die gesamte Rotation durchführt und dabei collect_bits passend aufruft. Die Startadresse der Quellbytes soll im Register EBX übergeben werden. Das Ergebnis soll in 8 aufeinanderfolgende Bytes geschrieben werden, deren Startadresse im Register ESI übergeben wird. Es dürfen alle Register verändert werden.

Hinweis: Für den Aufruf von collect_bits gelten folgende Konventionen: Die Startadresse der 8 Bytes wird in EBX erwartet, die zu sammelnde Bitposition in ECX. Das Ergebnis wird in AL zurückgegeben. EBX und ECX sind nach der Ausführung von collect_bits unverändert.

2.2.2 Sammeln einer Spalte

Ergänzen Sie die Lücken im folgenden 80386-Unterprogramm `collect_bits`; beachten Sie dabei die Kommentare. Die Startadresse der 8 Bytes stehe im Register `EBX`, die zu sammelnde Bitposition `n` im Register `ECX`. Das Ergebnis soll im Register `AL` stehen. `EBX` und `ECX` sollen nach der Ausführung nicht verändert sein.

`collect_bits :`

_____ `ebx`

`mov eax,0`

`mov edx,_____ ; 8 Bytes durchlaufen`

`collect_loop :`

`mov ah,_____ ; Byte holen`

`sh_____ ah,_____ ; Bit #n jetzt in Bit 0`

`sh_____ ah,_____ ; Bit #n jetzt im Carryflag`

`_____ al,1 ; Vom CF nach Bit 7 von AL`

`_____ ebx ; Nächstes Byte`

`_____ edx ; Schleifenzaehler`

`j_____`

Aufgabe 3 - Mikroprogrammierung

In der Anlage finden Sie ein Merkblatt mit einer Kurzbeschreibung der mikroprogrammierbaren Maschine, in der Sie alle zur Lösung der Aufgabe notwendigen Angaben finden, z.B. die Beschreibung des Mikroinstruktionsformats und die Funktionstabellen der Bausteine.

Das Mikroprogramm IFETCH, abgelegt ab Adresse 0x000 des MikroprogrammSpeichers, kann als gegeben betrachtet werden. Es holt den nächsten Maschinenbefehl aus dem Hauptspeicher in das Instruktionsregister, inkrementiert den Befehlszähler und springt das dem Befehls-Opcode zugehörige Mikroprogramm an. IFETCH benötigt 3 Takte zur Ausführung. Mikroprogramme müssen am Ende wieder zum Anfang des Mikroprogramms IFETCH zurückspringen. Hexadezimale Zahlen werden in Java- bzw. C-Schreibweise dargestellt und beginnen mit der Buchstabenkombination 0x, z.B: 0x1234, 0x011A oder 0xBEEF.

Gegeben sind folgende Maschinenbefehle:

| Opc. | Maschinenbefehl | Beschreibung |
|------|----------------------------|---|
| 0x02 | MOV <i>imm</i> , RB | Kopiert den Wert <i>imm</i> in das Register RB . |
| 0x03 | MOV RA , RB | Kopiert den Wert des Registers RA in das Register RB . |
| 0x12 | INC RB | Inkrementiert RB um 1 ($\mathbf{RB} = \mathbf{RB} + 1$). Die Flags des Maschinenstatusregisters werden entsprechend des Ergebnisses gesetzt. |
| 0x13 | DEC RB | Dekrementiert RB um 1 ($\mathbf{RB} = \mathbf{RB} - 1$). Die Flags des Maschinenstatusregisters werden entsprechend des Ergebnisses gesetzt. |
| 0x22 | JC <i>adr</i> | Bedingter Sprung nach <i>adr</i> , wenn das Carry-Flag im Maschinenstatusregister gesetzt ist. |
| 0x33 | JNZ <i>adr</i> | Bedingter Sprung nach <i>adr</i> , wenn das Zero-Flag im Maschinenstatusregister nicht gesetzt ist. |

| Opc. | Maschinenbefehl | Beschreibung |
|------|------------------------------|---|
| 0x42 | CMP [RA+imm1], [RB+imm2] | Führt die Subtraktion von [RA+imm1] minus [RB+imm2] durch. Dabei bezeichnet [Rx+imm] den Wert, der an der Adresse Rx+imm im Hauptspeicher liegt. Die Flags des Maschinenstatusregisters werden entsprechend des Ergebnisses gesetzt. Das Ergebnis wird verworfen. |
| 0x53 | SWAP [RA+imm1], [RB+imm2] | Führt eine Vertauschung der Werte durch, die im Hauptspeicher an den durch RA+imm1 und RB+imm2 gegebenen Adressen gespeichert sind. |

Anmerkung: Die Bezeichnung "RA" (bzw. "RB") steht hier abkürzend für "das durch das A-Registeradressfeld (bzw. B-Registeradressfeld) des Maschinenbefehlswortes adressierte Register".

3.1 Welche der oben angegebenen Befehle belegen

a) 16 Bit

b) 32 Bit

c) 48 Bit

im Hauptspeicher? Bitte die jeweiligen Opcodes angeben!

3.2 Wieviele Speicherzugriffe müssen von den Implementierungen folgender Maschinenbefehlen mindestens durchgeführt werden, wenn IFETCH nicht beachtet wird?

a) MOV imm, RB

b) INC RB

c) CMP [RA+imm1], [RB+imm2]

d) SWAP [RA+imm1], [RB+imm2]

3.3 Gegeben ist folgendes Assemblerprogramm, das nur die obigen Maschinenbefehle nutzt:

```
schleife1: MOV    r0, r2
           MOV    r1, r3
           DEC    r3
           MOV    1, r4
schleife2: CMP    [r2 + 1], [r2]
           JC     weiter
           SWAP  [r2], [r2 + 1]
           INC    r4
weiter:    INC    r2
           DEC    r3
           JNE   schleife2
           DEC    r4
           JNE   schleife1
```

3.3.1 Welche Bedingung muss für die CMP-Operanden $A = [r2+1]$ und $B = [r2]$ im Assemblerprogramm vorliegen, damit JC einen Sprung ausführt? Welche Aufgabe erfüllen also die Instruktionen zwischen "schleife2:" und "weiter:"?

3.3.2 Es sei der folgende Hauptspeicherinhalt gegeben:

| Adresse | 0x5000 | 0x5001 | 0x5002 | 0x5003 |
|---------|--------|--------|--------|--------|
| Wert | 1 | 5 | 4 | 2 |

Das Assemblerprogramm wird mit Anfangsbelegung $r0=0x5000$, $r1=4$ aufgerufen. Welche Belegung haben die Speicherzellen $0x5000$ bis $0x5003$, wenn `DEC r4` das erste Mal ausgeführt wird?

3.3.3 Welche Aufgabe erfüllt das Assemblerprogramm?

3.3.4 Zeigen Sie, wie der innere Teil des Assemblerprogramms in hexadezimaler Codierung aussieht.

| Adresse | Inhalt | Befehl |
|---------|--------|-------------------------------|
| ... | | |
| 0x0103 | _____ | MOV 1, r4 |
| _____ | _____ | |
| _____ | _____ | |
| _____ | _____ | schleife2: CMP [r2 + 1], [r2] |
| _____ | _____ | |
| _____ | _____ | JC weiter |
| _____ | _____ | |
| _____ | _____ | SWAP [r2], [r2 + 1] |
| _____ | _____ | |
| _____ | _____ | INC r4 |
| _____ | _____ | |
| _____ | _____ | weiter: INC r2 |
| _____ | _____ | |

3.4 Folgende Maschinenbefehle sollen nun durch ein Mikroprogramm durch Vervollständigen der Tabellen auf Seite 14/15 realisiert werden:

- DEC RB
- CMP [RA + *imm1*], [RB + *imm2*]

Die Tabelle enthält Reservezeilen für einen zweiten Lösungsversuch, bitte streichen Sie falsche Lösungen deutlich durch!

Bitte keine binären Werte eintragen, sondern die Abkürzungen aus dem Merkblatt MI (in der Anlage) verwenden.

_____ Tabelle auf der nächsten Seite! _____

| | 79 | 78 | 77 | 76 | 75 | 74 | 73 | 72 | 71 | 70 | 69 | 68 | 67 | 66 | 65 | 64 | 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | | |
|-------------------------|-----------|----|----|----|-----------|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|-----|------|------|---------|----|----|---------|----|----|----|----|----|----|----|------|----|----|----|----|------|----|--|--|
| IE | I3 | I2 | I1 | I0 | KMUX | K15 | K14 | K13 | K12 | K11 | K10 | K9 | K8 | K7 | K6 | K5 | K4 | K3 | K2 | K1 | K0 | I2 | I1 | I0 | I5 | I4 | I3 | I8 | I7 | I6 | A3 | A2 | A1 | A0 | ASEL | B3 | B2 | B1 | B0 | BSEL | | | |
| | Interrupt | | | | Konstante | | | | | | | | | | | | | | | | Src | Func | Dest | RA Addr | | | RB Addr | | | | | | | | | | | | | | | | |
| Adr. DEC RB | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 130 | DIS | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CMP [RA+imm1],[RB+imm2] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 420 | DIS | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 421 | DIS | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 422 | DIS | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 423 | DIS | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 424 | DIS | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 425 | DIS | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 426 | DIS | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 427 | DIS | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------------------|-----|---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| Adr. DEC RB | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 130 | DIS | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CMP [RA+imm1],[RB+imm2] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 420 | DIS | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 421 | DIS | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 422 | DIS | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 423 | DIS | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 424 | DIS | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 425 | DIS | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 426 | DIS | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 427 | DIS | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------|------------|-----------------------|-----|----|------------------------|----|----|--------|------|-------------------|-----|----|--------|----|-------------|-------|----|----|----|----|-----|-----|----|----|----|----|----|----|----|----|----|----|---------|--------|---------|--------|--------|------|
| 38 | 37 | 36 | 35 | 34 | 33 | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ABUS* | DBUS* | I12 | I11 | I9 | I8 | I7 | I6 | CEMUE* | CEM* | I5 | I4 | I3 | I2 | I1 | I0 | CCEN* | I3 | I2 | I1 | I0 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | BZ_LJD* | BZ_ED* | BZ_INC* | BZ_FA* | IR_LD* | MWE* |
| Y- MUX | CIN MUX | Schiebe- steuerung | | | Statusregister Test | | | | | AM2910 Befehle | | | | | Direktdaten | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | | | | | | | | P S | CJP | | IFETCH | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | | | H | H | X | X | CONT | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | | | H | H | X | X | CONT | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | | | H | H | X | X | CONT | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | | | H | H | X | X | CONT | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | | | H | H | X | X | CONT | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | | | H | H | X | X | CONT | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | | | | | | | | P S | CJP | | IFETCH | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|---|--|--|---|---|---|---|------|--------|-----|--|--------|--|--|--|--|--|--|--|--|--|--|--|--|
| | | X | | | | | | | | P S | CJP | | IFETCH | | | | | | | | | | | | |
| | | X | | | H | H | X | X | CONT | X | | | | | | | | | | | | | | | |
| | | X | | | H | H | X | X | CONT | X | | | | | | | | | | | | | | | |
| | | X | | | H | H | X | X | CONT | X | | | | | | | | | | | | | | | |
| | | X | | | H | H | X | X | CONT | X | | | | | | | | | | | | | | | |
| | | X | | | H | H | X | X | CONT | X | | | | | | | | | | | | | | | |
| | | X | | | H | H | X | X | CONT | X | | | | | | | | | | | | | | | |
| | | X | | | | | | | | P S | CJP | | IFETCH | | | | | | | | | | | | |

Aufgabe 4 - Rechnergestützter Schaltungsentwurf

4.1 Parallele Parityberechnung

Im Folgenden soll ein Schaltkreis zur Berechnung der sog. *Parität* eines 4Bit-Werts entwickelt werden. Die Parität (engl. parity) ist ein einzelnes Bit, das 1 ergibt, wenn die Anzahl der 1er-Bits gerade ist. Ansonsten (ungerade Anzahl) ist es 0.

Die vier Eingangsbits sind a,b,c und d, das Paritybit p.

4.1.1 Ergänzen Sie die Wertetabelle für die Paritätsberechnung.

| a | b | c | d | p |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | |
| 0 | 0 | 0 | 1 | |
| 0 | 0 | 1 | 0 | |
| 0 | 0 | 1 | 1 | |
| 0 | 1 | 0 | 0 | |
| 0 | 1 | 0 | 1 | |
| 0 | 1 | 1 | 0 | |
| 0 | 1 | 1 | 1 | |
| 1 | 0 | 0 | 0 | |
| 1 | 0 | 0 | 1 | |
| 1 | 0 | 1 | 0 | |
| 1 | 0 | 1 | 1 | |
| 1 | 1 | 0 | 0 | |
| 1 | 1 | 0 | 1 | |
| 1 | 1 | 1 | 0 | |
| 1 | 1 | 1 | 1 | |

4.1.2 Erstellen Sie die disjunktive Normalform (DNF) für p.

4.1.3 Erstellen Sie die konjunktive Normalform (KNF) für p.

4.1.4 Wie lautet der boolsche Ausdruck für p, wenn alle boolschen Operationen erlaubt sind?

4.1.5 Erstellen Sie eine VHDL-Architecture, die die Parityberechnung implementiert. Als Eingänge sind wieder a,b,c und d definiert, das Ausgangssignal soll p heißen. Alle Signale sind vom Typ std_logic.

4.2 Sequentielle Parityberechnung

Neben der oben beschriebenen parallelen Berechnung des Paritywerts ist es auch möglich, ihn schrittweise, also Bit für Bit zu berechnen. Das wird dann interessant, wenn die Daten ohnehin nur seriell eintreffen. Es soll nun ein VHDL-Baustein entwickelt werden, der für 32 aufeinanderfolgende Bits das Paritybit berechnet. Dabei kommt mit jeder steigenden Taktflanke an CLK ein einzelnes Bit am Eingang D an.

Trotz des kontinuierlichen Bitstroms soll die Parityberechnung aber erst beginnen, wenn ein Startsignal START für einen Takt auf 1 gesetzt wurde. Die Berechnung beginnt im darauffolgenden Takt.

Sind die 32 Bits verarbeitet worden, soll das Ergebnis auf den Ausgang P gelegt werden. Um zu signalisieren, dass das Ergebnis gültig ist, soll für einen Takt das Ausgangssignal P_VALID auf den Wert 1 gesetzt werden (sonst 0).

4.2.1 Erstellen Sie die VHDL-Entity des Bausteins. Halten Sie sich an die vorgegebenen Signalnamen.

4.2.2 Mit welcher Berechnung kann man den neuen Paritywert aus dem alten Wert und dem D-Eingang bestimmen?

$p_{neu} = \underline{\hspace{10em}}$

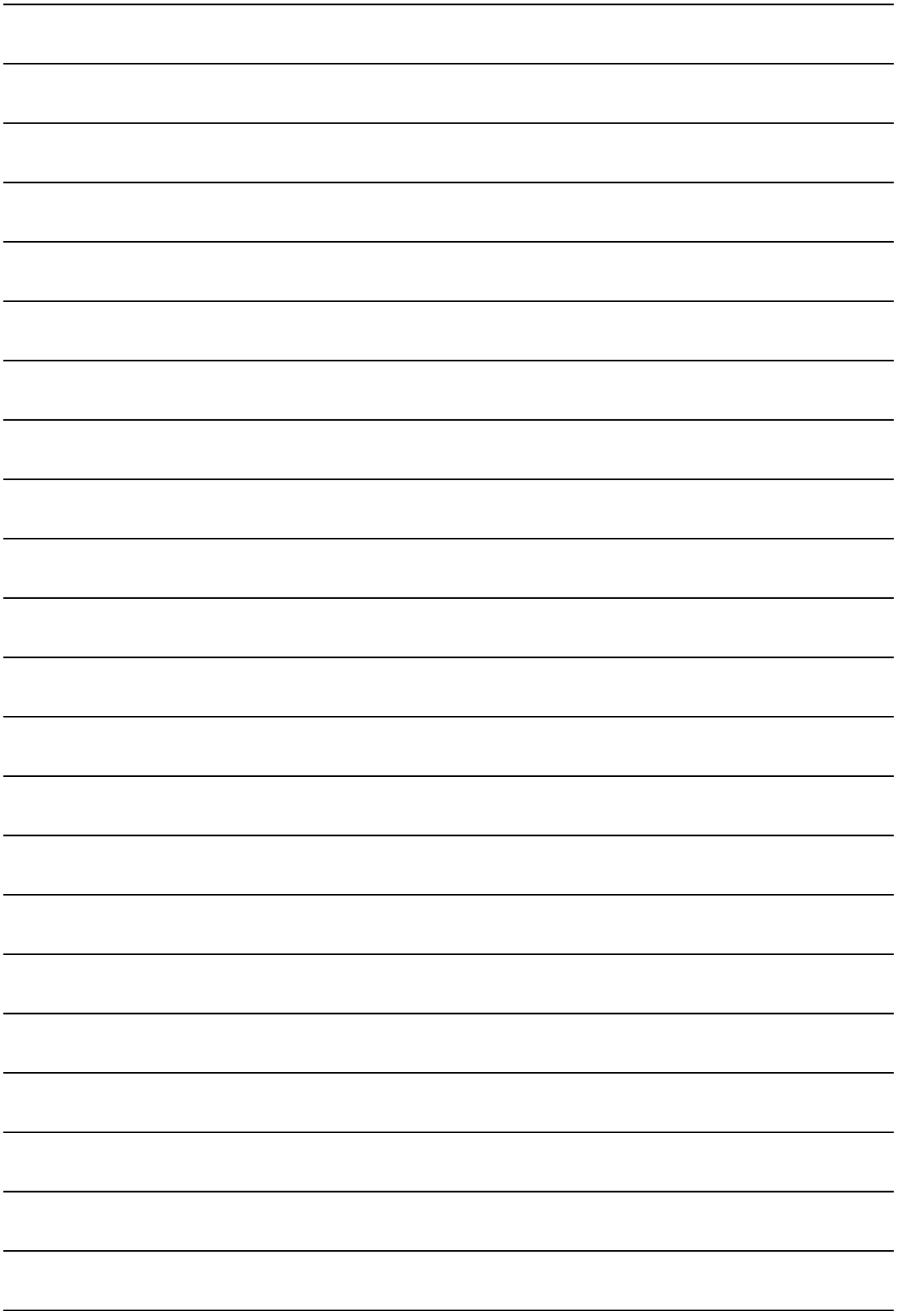
4.2.3 Wieviele interne Zustände hat der Baustein mindestens?

4.2.4 Zeichnen Sie das Zustandsübergangsdiagramm (“Blasendiagramm”) des Bausteins. Kennzeichnen Sie auch die Ausgabe von P_VALID.

Hinweis: Da viele ähnliche Zustände und Übergänge vorhanden sind, ist es ausreichend, wenn Sie nur Anfang und Ende der Übergangskette zeichnen.



4.2.5 Erstellen Sie die VHDL-Architecture des Bausteins.



Lösung für Aufgabe ...

Lösung für Aufgabe ...
