

SIMM-Tester

Ein universelles Testgerät für Speicher-SIMMs

Gebaut als Praktikum an der
Abteilung für Systemsoftware
im Sommersemester 1999

Verfasser:
Manfred Meindl
9055664 / 881

Betreuer:
Prof. Dr. H. Mössenböck

Technische Betreuung:
Ing. Anton Kral

1	GRUNDSÄTZLICHE AUFGABENSTELLUNG	3
2	SPEZIFIKATION	3
2.1	ANSCHLUß	3
2.2	BENUTZER-SCHNITTSTELLE.....	3
2.3	TESTBARE SIMMS	3
2.4	TESTERGEBNISSE	4
3	LÖSUNG	4
3.1	ERSTER ANSATZ	4
3.2	AUFTRETENDE PROBLEME UND LÖSUNGEN	4
4	VERWENDETE RESSOURCEN DES MIKROCONTROLLERS.....	5
4.1	VERWENDETE PORTS.....	5
4.2	VERWENDETE TIMER.....	5
5	SOFTWARE-STRUKTUR.....	5
5.1	HAUPTSCHLEIFE.....	6
5.2	ZUSTANDSAUTOMAT	6
5.3	EREIGNIS-WARTESCHLANGE	8
5.4	WARTESCHLANGE FÜR LCD-KOMMANDOS	8
5.5	REFRESH-EINHEIT	8
5.6	SCAN-ALGORITHMUS	8
5.7	TEST-ALGORITHMUS	8
6	HARDWARE-STRUKTUR.....	8
6.1	GLIEDERUNG	8
6.2	DARSTELLUNGEN.....	9
7	KONFIGURATION DES ENTWICKLUNGSBOARDS.....	9
8	BEDIENUNGSANLEITUNG	9
8.1	INBETRIEBNAHME	9
8.2	MENÜ-FÜHRUNG.....	9
8.3	SIMM TESTEN.....	10
9	UNERWARTETE PROBLEME UND IHRE LÖSUNG	10
9.1	TECHNOLOGISCHE PROBLEME MIT DER PLATINE	10
9.2	PROBLEME BEI DER INITIALISIERUNG DES LCD-DISPLAYS	10
9.3	FEHLERHAFT FUNKTION DER SIMMS DURCH "POWER- UND GROUND-BOUNCE"	10
10	VERWENDETE HILFSMITTEL	11
10.1	VERWENDETE HARDWARE-KOMPONENTEN.....	11
10.2	VERWENDETE KOMPONENTEN UND SOFTWARE-TOOLS.....	11
10.3	VERWENDETE LITERATUR	11
11	ANHANG A: SCHALTPLAN.....	12
12	ANHANG B: PLATINEN-LAYOUT.....	13
13	ANHANG C: BESTÜCKUNGSPLAN	14
14	ANHANG D: STÜCKLISTE	15
15	ANHANG E: DATEIEN AUF DER CD	16

1 Grundsätzliche Aufgabenstellung

Zu entwickeln ist ein Testgerät, das die für PCs und ähnliche Geräte üblichen Speicherkarten (Single Inline Memory Modules oder SIMMs) erkennen, klassifizieren und auf Fehler überprüfen kann.

Dabei soll jedoch nicht die Zugriffszeit (meist nur wenige ns) bestimmt werden. Dies wäre mit den zur Verfügung stehenden Mitteln nicht zu erreichen. Vielmehr soll nur die logische Funktion der Bausteine untersucht werden.

Die Prüfung soll durch einfache Bedienung weniger Tasten mit Rückkopplung über eine Anzeige weitgehend intuitiv gesteuert werden können.

Die Ergebnisse der Prüfung sind auf einer Anzeige darzustellen.

Die aufzubauende Schaltung soll von einem Mikrocontroller-Entwicklungsboard für den Mikrocontroller Siemens C167CR-LM gesteuert werden. Dieser ist dafür entsprechend zu programmieren.

2 Spezifikation

2.1 Anschluß

Anschluß an das Entwicklungsboard EVA167 der Firma Ertec und eine Stromversorgung mit Masse und +7V bis 9V(maximal 1A). Über diese Stromversorgung muß auch das Entwicklungsboard versorgt werden.

2.2 Benutzer-Schnittstelle

Die Anzeige der Statusinformationen erfolgt auf einem alphanumerischen LCD-Punktmatrix-Modul mit 2 Zeilen und 16 Zeichen pro Zeile. Das Anzeige-Modul hat einen eigenen Displaycontroller.

Der Benutzer kann über 4 Tasten den Testvorgang steuern.

Der Testvorgang kann vom Bediener des SIMM-Testers gestartet und auch vorzeitig beendet werden. Ergebnisse können abgefragt werden.

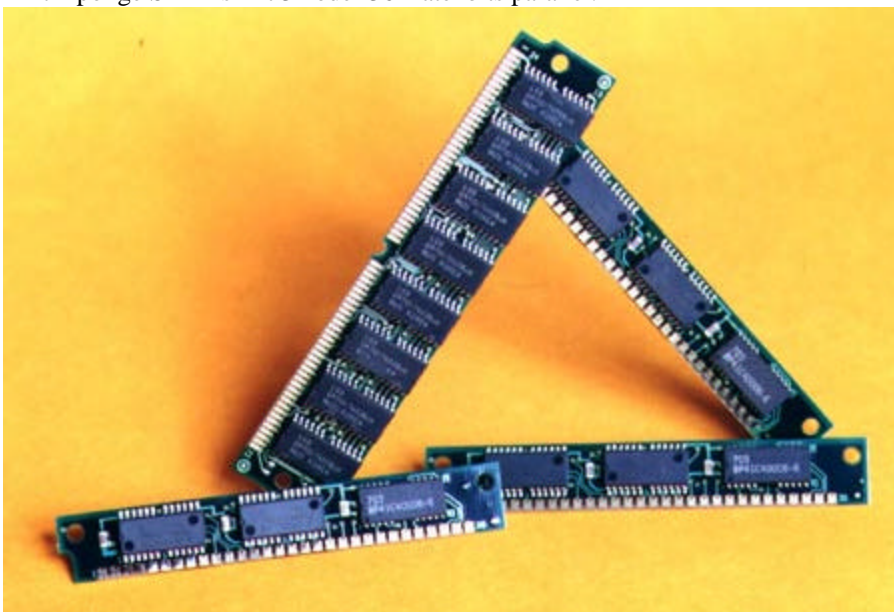
2.3 Testbare SIMMs

Es kann immer nur ein SIMM gleichzeitig eingelegt und getestet werden.

2.3.1 Testbare Bauformen von SIMMs:

Folgende Bauformen werden unterstützt:

- 30-polige SIMMs mit 8 oder 9 Datenbits parallel.
- 72-polige SIMMs mit 32 oder 36 Datenbits parallel.



Die heute oft anzutreffenden DIMMs (Dual Inline Memory Modules) mit 168 Anschlüssen wurden nicht berücksichtigt, da dies nur mit einer professionell hergestellten Platine zu erreichen wäre.

2.3.2 Testbare Speichergrößen:

Zum Zeitpunkt des Projektstarts waren die größten verfügbaren SIMMs solche mit 8 Megaworten zu je 36 Bits. Diese können getestet werden.

Wenn größere SIMMs spezifiziert werden, können sie vermutlich ohne Änderung der Hard- oder Software gleich getestet werden. Ansonsten kann durch ein Software-Update für unseren Mikrocontroller auch deren Testbarkeit erreicht werden. Die Hardware ist bereits darauf vorbereitet bzw. kann leicht adaptiert werden.

2.3.3 Testbare Speichertechnologien:

Bezüglich der unterstützten Speichertechnologien gilt:

- Die Versorgungsspannung muß 5 Volt betragen. Dies ist die einzige übliche Spannung für SIMMs. Geringere Versorgungsspannungen von z.B. 3 Volt gibt es nur bei DIMMs.
- Unterstützt werden auch Speicher für schnelle Zugriffsmodi wie FPM und EDO.
- Unterstützt werden Speicher mit Refreshmodus RAS#-only. Dieser wird von praktisch allen dynamischen RAMs akzeptiert.

2.4 Testergebnisse

Die Ergebnisse des Tests müssen folgendes umfassen:

- Die Größe des SIMMs (wieviel Speicherstellen zu je wieviel Bits)
- Ist der SIMM OK oder defekt?
- Wenn er defekt ist, welche Bits sind von Fehlern betroffen.

3 Lösung

3.1 Erster Ansatz

Die erste Idee zur Lösung der Aufgabe ist trivial: Die parallelen Ports des Mikrocontrollers, die am Entwicklungsboard zu Verfügung stehen, werden mit sämtlichen Kontakten der zu testenden SIMMs verbunden. Dann werden durch entsprechende Signale Daten im SIMM abgelegt und wieder zurückgelesen. Durch Vergleich der Daten kann auf die Tauglichkeit des SIMM geschlossen werden.

Zusätzlich ist noch die Mensch-Maschine-Schnittstelle mit Tastern und der LCD-Anzeige zu überwachen bzw. zu steuern, was wiederum mit parallelen Ports am einfachsten zu bewerkstelligen ist.

3.2 Auftretende Probleme und Lösungen

3.2.1 Zu wenige parallele Ports am Mikrocontroller verfügbar

Dieses Problem tritt vor allem bei den 72-poligen SIMMs auf. Diese heute oftmals verwendeten SIMMs haben neben den mehrfach vorhandenen Versorgungsleitungen alleine bis zu 36 Anschlüsse für Datenbits. Dazu kommen noch bis zu 12 Adressleitungen und eine Vielzahl von Steuer- und Statusleitungen.

In Summe werden also ca. 60 Anschlüsse für Logiksignale benötigt. Die parallelen Ports, die der verwendete Mikrocontroller bietet, reichen nicht aus, um alle diese Signale zu steuern und zu lesen.

Die Lösung des Problems heißt (De)Multiplexing. Dabei wird einerseits der Datenbus der SIMMs in 4 Teile geteilt. Vier Teile zu je maximal 9 Datenbits, die parallelgeschaltet sind und nur durch verschiedene Steuerleitungen aktiviert werden.

Dadurch spart man immerhin 27 (= 9*3) Pins.

Beim Schreiben in den Speicher kann dann das selbe Muster gleichzeitig in alle 4 Teile geschrieben werden. Beim Zurücklesen muß aber jeder Teil einzeln gelesen und verglichen werden, um Konflikte am Bus zu vermeiden. Dies reduziert die Geschwindigkeit des Testers ein wenig, ist aber nicht zu vermeiden.

Andererseits wird der Adreßbus des Speichers nicht direkt angesprochen, sondern über ein Latch. Dadurch kann der schon eingeplante Datenbus auch für die Adressen verwendet werden. Da der Adreßbus 12 Bit breit ist, ergibt sich nach Abzug zweier Steuerleitungen für das Latch eine Einsparung von 10 Pins.

Durch diese Maßnahmen reichen die parallelen Ports des Mikrocontrollers aus und bieten auch noch Platz für den Anschluß von LCD-Display und Tastern sowie die Möglichkeit für spätere Erweiterungen.

3.2.2 Refresh der SIMMs

Wenn auf eine Speicherzelle in einem dynamischen RAM längere Zeit (typisch 4 bis 64 ms, also eigentlich gar nicht so lang) nicht zugegriffen wird, geht der Speicherinhalt verloren. Dies kann durch sogenannte Refresh-Zyklen verhindert werden. Dabei werden die Information tragenden Ladungen wieder aufgefrischt.

Da dies regelmäßig geschehen muß, wird im Mikrocontroller ein Timer aktiviert, der in Abständen von ca. 15.6µs einen Interrupt auslöst. Die entsprechende Interrupt-Serviceroutine führt dann einen Refresh-Zyklus einer Speicherzeile durch.

Die Hardware muß hierfür nicht erweitert werden, da der Refresh über die normalen Steuerleitungen des Speichers gesteuert werden kann.

Durch das Verwenden der gleichen Leitungen für Speicherzugriffe und Refresh sind Konflikte auf diesen Leitungen vorprogrammiert. Dies muß durch die Software verhindert werden.

In der Praxis braucht man beim Testen der SIMMs aber kein Refresh, weil durch das ständige schnelle und gleichmäßig verteilte Zugreifen auf den Speicher die Speicherzeilen ohnehin ausreichend oft aufgefrischt werden. Auch kostet der Refresh ziemlich viel CPU-Leistung, weshalb er aus Gründen der Performanz während des Testens nicht verwendet wird. Nur wenn die Testsequenz unterbrochen wird, wird während dieser Unterbrechung ein regelmäßiger Refresh durchgeführt.

3.2.3 Interaktion mit dem Benutzer

Statusinformationen wie die Größe des SIMMs und der Testfortschritt werden dem Benutzer auf einer kleinen alphanumerischen LCD-Punktmatrix-Anzeige angezeigt. Der Datenbus des Displays (4 Bit breit; 8 Bit wäre auch möglich) wird mit dem Mikrocontroller-Entwicklungsboard verbunden. Ebenso die Steuerleitungen des Displays. Steuerung des Tests ist über Tasten möglich. Dazu werden von den Tasten externe Interrupts am Mikrocontroller ausgelöst.

Diese Interaktion muß reaktiv sein. Das heißt, sie muß schnell und unabhängig von laufenden Tests funktionieren. Um dies zu erreichen werden Warteschlangen und Interrupts verwendet.

4 Verwendete Ressourcen des Mikrocontrollers

4.1 Verwendete Ports

Folgende Ports des Mikrocontrollers werden verwendet:

- Port 2: Wird für Steuersignale der SIMMs (RAS, CAS, WE), Steuersignale der Latches (Latch enable, Output enable) und das Statussignal vom 72-poligen SIMM (PD) verwendet.
- Port 3: Wird für Adress- und Datenbus der SIMMs (12 bzw. 9 Bit) verwendet.
- Port 7: Wird für die Ansteuerung des LCD-Displays verwendet.
- Port 8: Wird für die Tasten verwendet. Jede Taste löst beim Drücken einen Interrupt aus.

4.2 Verwendete Timer

Folgende Timer und deren Überlauf-Interrupts werden verwendet:

- Timer 0 für das Entprellen der Tasten und die Wartezeiten beim Initialisieren des Displays.
- Timer 1 für permanentes Scrollen beim Lesen des Info-Textes.
- Timer 7 für den regelmäßigen Speicher-Refresh.
- Timer 8 für das Schreiben auf die LCD-Anzeige.

5 Software-Struktur

Die gesamte Software ist darauf ausgerichtet, den System-Durchsatz zu maximieren. Dies ist sinnvoll, da das eigentliche Testen der SIMMs vor allem bei großen Exemplaren recht zeitaufwendig ist und daher möglichst wenig verzögert werden sollte.

Die ganze Software ist ereignisorientiert. Das heißt, Ereignisse lösen Aktionen aus, die wiederum Ereignisse generieren.

5.1 Hauptschleife

Das Hauptprogramm läuft nach der Initialisierung des gesamten Systems in einer Endlosschleife. In dieser Schleife wird:

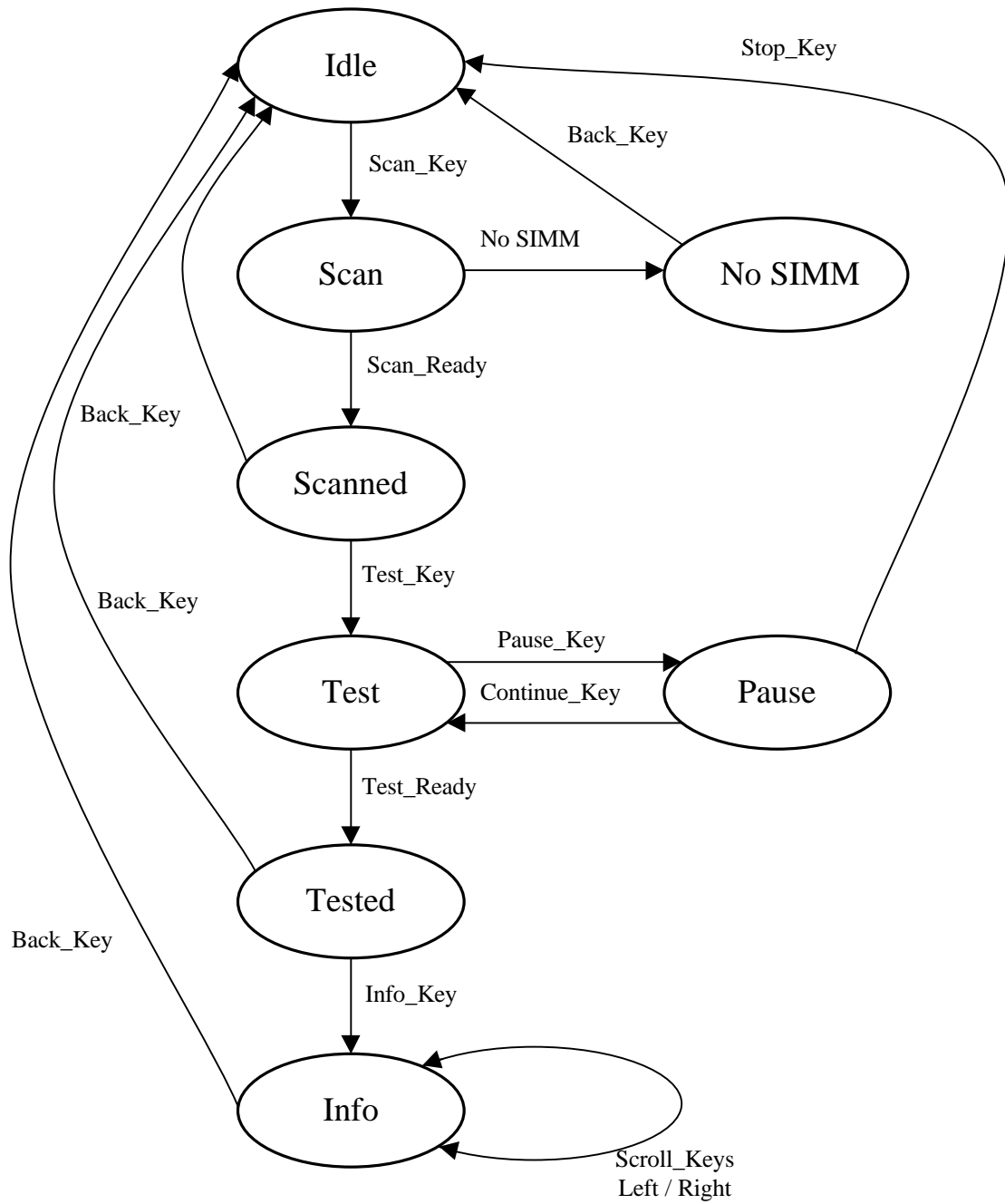
- Auf anstehende Ereignisse (Events) hin geprüft und deren Verarbeitung angestoßen.
- Bei Bedarf den SIMM scannen. Also seine Größe bestimmen.
- Bei Bedarf ein Teil des SIMMs getestet.
- Wenn weder Ereignisse zu verarbeiten noch Seitentests zu machen sind, fungiert die Hauptschleife als Idle-Schleife.

5.2 Zustandsautomat

Die Kontrolle des Ablaufes einer SIMM-Prüfung wird vom Benutzer in Kooperation mit einem endlichen Automaten (state machine) übernommen. Letzterer übernimmt von der Hauptprogramm-Schleife die aktuellen Ereignisse und wechselt dadurch eventuell den Zustand des Automaten und somit des ganzen Programmes. Das Zustandsdiagramm ist umseitig dargestellt.

Die Tastenbelegung ist dabei wie folgt:

- Scan_Key, Test_Key, Info_Key, Continue_Key: Rechte Taste.
- Pause_Key, Back_Key, Stop_Key: Linke Taste.
- Scroll_Keys: Die beiden mittleren Tasten.



5.3 Ereignis-Warteschlange

Anfallende Ereignisse (Events) werden in einer FIFO-Warteschlange verwaltet und vom Hauptprogramm verarbeitet. Dadurch können Ereignisse sehr schnell hintereinander auftreten, ohne sich gegenseitig zu blockieren. Ereignisse sind dabei

- das Drücken einer Taste über Interrupts.
- das Gedrückthalten einer Scroll-Taste beim Lesen des Info-Textes führt regelmäßig zu einem Event.
- das Beenden des Scanvorganges.
- das Beenden des Testes des gesamten SIMMs von der Hauptschleife aus.

Geleert wird die Warteschlange von der Hauptschleife.

5.4 Warteschlange für LCD-Kommandos

Anfallende Kommandos und Daten für die LCD-Anzeige werden in einer FIFO-Warteschlange verwaltet und von einer Timer-Interruptroutine verarbeitet. Dadurch kann die relativ langsame Anzeige effizient genutzt werden, ohne daß auf sie gewartet werden muß. Für noch höhere Effizienz wird die Länge des Timerintervalls bis zum nächsten Interrupt je nach dem letzten LCD-Kommando speziell eingestellt. Dadurch werden größere Datenmengen schnell verarbeitet, ohne die CPU durch zu viele Interrupts zu belasten. Das führt zu geringerem Flimmern der Anzeige bei gleichzeitiger hoher Testgeschwindigkeit.

5.5 Refresh-Einheit

Ein Timer-Interrupt aktiviert in regelmäßigen Abständen (15.6 μ s) die Refresh-Einheit des Systems, um den Speicherinhalt im SIMM aufzufrischen. Dies wird in der aktuellen Implementierung während des Testens nicht mehr gemacht, da durch das Testen selbst der Speicherinhalt ständig aufgefrischt wird. Refresh findet nur mehr bei unterbrochenem Test statt.

5.6 Scan-Algorithmus

Dabei wird die Größe des SIMMs ermittelt. Folgende Vorgangsweise hat sich als praktikabel erwiesen:

- Zunächst wird festgestellt, wieviele Bits der Speicher parallel bietet (8, 9, 32 oder 36).
- Dann werden die Zeilen- und Spaltenzahl ermittelt.
- Durch Multiplizieren von Zeilen- und Spaltenzahl erhält man die Größe des SIMMs in Worten.

Das Problem dabei ist, das der Scan-Vorgang auch bei defekten SIMMs richtige Ergebnisse liefern soll. Um dies zu erreichen, werden hier keine strengen Vergleiche zwischen geschriebenen und zurückgelesenen Werten angestellt. Vielmehr wird die Hamming-Distanz (also die Zahl der unterschiedlichen Bits) dieser Werte berechnet. Liegt diese unter 3, so gilt der Wert als korrekt geschrieben und gelesen. Daraus ergibt sich auch, daß maximal 3 Bits fehlerhaft sein dürfen, damit der SIMM überhaupt erkannt wird.

Wenn Adressleitungen defekt sind, kann der SIMM eventuell überhaupt nicht oder zumindest nicht richtig erkannt werden.

5.7 Test-Algorithmus

Der Algorithmus, der zum Testen der SIMMs verwendet wird, gliedert sich in drei Teile. Hier wird zur Vereinfachung von einem 30-poligen SIMM ohne Paritätsbit ausgegangen:

- Der Speicher wird vollständig (mit steigenden Adressen) mit dem Bytewert 0x55 gefüllt.
- Mit fallenden Adressen wird nun für jede Adresse getestet, ob der Wert 0x55 auch zurückgelesen werden kann. Außerdem wird als neuer Wert 0xAA (also das Inverse von 0x55) geschrieben.
- Mit steigenden Adressen wird nun getestet, ob der neue Wert 0xAA auch zurückgelesen werden kann. Außerdem wird wieder der erste Wert 0x55 in alle Speicherzellen geschrieben.

6 Hardware-Struktur

6.1 Gliederung

Die Hardware gliedert sich in drei Teile:

- Die Schnittstelle zum Mikrocontroller-Entwicklungsboard
- Die Schnittstelle zu den zu testenden SIMMs
- Das Benutzer-Interface

6.2 Darstellungen

Der Schaltplan sowie die Layouts und Bestückungspläne und Stückliste der Platine sind in den Anhängen A bis D dargestellt.

Der Schaltplan ist leicht zu verstehen. Hier noch einige Hinweise zu besserer Orientierung:

- Im Schaltplan liegt die Schnittstelle zum Entwicklungsboard ganz links.
- Die Sockel für die SIMMs sind rechts dargestellt. Die nötigen Latches liegen im oberen Bereich der Zeichnung.
- Das Benutzer-Interface ist links unten dargestellt.

7 Konfiguration des Entwicklungsboards

Für die Steuerung des SIMM-Testers wird das Entwicklungsboard EVA167 der Firma Ertec verwendet. Dieses ist mit einem Mikrocontroller vom Typ C167CR-LM (ohne internes ROM oder Flash) bestückt und verfügt in der verwendeten Bestückung über 256KB Hauptspeicher. Es wurde wie folgt konfiguriert:

- Taktfrequenz = 20MHz durch Verwendung eines Oszillators mit 5 MHz und Vervierfachung des Taktes durch die im Mikrocontroller integrierte PLL.
- Zugriff auf externen Speicher über einen 16 Bit breiten gemultiplexten Adress- und Datenbus und zwei weitere Adressleitungen.
- Aktivierter Bootstrap-Modus zur Verwendung mit dem Monitorprogramm auf einem Host-PC. Der Host-PC ist dabei über eine 9-polige serielle Verbindung mit dem Entwicklungsboard verbunden.
- Die Software des SIMM-Testers ist auch im Flash-Speicher des Entwicklungsboards gespeichert, sodaß ein Host-PC nicht notwendig ist und der SIMM-Tester als stand-alone-Gerät verwendet werden kann.

8 Bedienungsanleitung

8.1 Inbetriebnahme

Zum Betrieb braucht der SIMM-Tester einen Anschluß an das Entwicklungsboard EVA167 der Firma Ertec und an eine Stromversorgung mit Masse und +7V bis 9V(maximal 1A).

Da die SIMMs nicht bei angelegter Betriebsspannung eingesteckt oder herausgezogen werden dürfen, sollte der zu testende SIMM sinnvollerweise gleich vor dem Anschluß an das Netzteil in den SIMM-Tester eingesteckt werden.

8.2 Menü-Führung

Das zweizeilige Display gliedert sich in zwei Teile:

- In der oberen Zeile wird der Zustand des Gerätes und des eingesteckten SIMMs angezeigt.
- In der unteren Zeile wird angezeigt, welche Funktion die darunter befindlichen Tasten gerade haben. Meist sind nur die ganz linke und die ganz rechte Taste aktiv (oder auch nur eine einzige).

Die mit den Tasten wählbaren Funktionen sind:

- Scan: Damit wird der Scan-Vorgang des SIMMs gestartet. Dabei wird die SIMM-Größe ermittelt.
- Test: Damit wird der Test-Vorgang des SIMMs gestartet. Dabei wird der SIMM auf Fehler hin überprüft. Dieser Vorgang kann einige Zeit dauern. Daher wird der Fortschritt durch einen Balken in der oberen Zeile des Displays angezeigt.
- Pause: Damit wird der laufende Test-Vorgang unterbrochen.
- Continue: Damit wird der unterbrochene Test-Vorgang fortgesetzt.
- Info: Damit werden Informationen zum SIMM und seinen Fehlern angezeigt.
- < > : Mit den beiden symbolisierten Pfeiltasten kann durch die angezeigte Information gescrollt werden. Dabei scrollt der Text um ein Zeichen beim kurzen Drücken und scrollt dauernd beim Halten der Pfeiltasten.
- Back: Damit kann zum Anfang zurückgekehrt werden.
- Stop: Damit kann von einem abgebrochenen Test zum Anfang zurückgekehrt werden.

8.3 SIMM Testen

- Vor dem Einstecken des SIMMs unbedingt die Versorgungsspannung abschalten!
- Den zu testenden SIMM einstecken.
- Die Spannungsversorgung anschließen.
- Den Reset-Knopf am Entwicklungsboard drücken.
- Die rechte Taste drücken, um den Scan-Vorgang zu starten.
- Je nach angezeigtem Status und gewünschter Information weiter durch das Menü navigieren.
- Vor dem Herausziehen des SIMMs unbedingt die Versorgungsspannung abschalten!
- Den SIMM entfernen.

9 Unerwartete Probleme und ihre Lösung

9.1 Technologische Probleme mit der Platine

Durch die Herstellung der Platine in Eigenregie war das Ergebnis von bescheidener Qualität und recht fehlerbehaftet. Die Herstellung und Fehlerbeseitigung dauerten enorm lange und hätten die Investition in eine professionell hergestellte Platine (ca. öS 500.-) nachträglich betrachtet gerechtfertigt.

9.2 Probleme bei der Initialisierung des LCD-Displays

Anfangs gelang es nicht, irgendetwas auf dem Display anzuzeigen. Es stellte sich schließlich heraus, daß die Initialisierung nicht funktioniert hatte. Diese ist etwas schwierig, wenn das nur 4 Bit schmale Interface verwendet wird.

Die Lösung war, eine Initialisierungsroutine aus dem originalen Datenblatt des Displaycontrollers HD44780 für meine Zwecke anzupassen. Hier nun der endgültige (funktionierende) Algorithmus für alle, die ähnliche Probleme wie ich haben oder gar nicht erst bekommen möchten:

- Nach Anlegen der Versorgungsspannung 100 ms warten
- Steuerleitungen initialisieren: LCD_E=0; LCD_RW=0; LCD_RS=0;
- Mindestens 15 ms warten
- LCD_Data=0x3;
- Enable-Leitung pulsen: LCD_E=0; LCD_E=1; LCD_E=0;
- Mindestens 5 ms warten
- Enable-Leitung pulsen: LCD_E=0; LCD_E=1; LCD_E=0;
- Mindestens 1 ms warten
- Enable-Leitung pulsen: LCD_E=0; LCD_E=1; LCD_E=0;
- Mindestens 1 ms warten
- LCD_Data=0x2;
- Enable-Leitung pulsen: LCD_E=0; LCD_E=1; LCD_E=0;
- Mindestens 1 ms warten
- LCD_Data=0x8;
- Enable-Leitung pulsen: LCD_E=0; LCD_E=1; LCD_E=0;
- Mindestens 1 ms warten

Danach kann die Anzeige ganz normal verwendet werden, wobei wieder das Timing einzuhalten ist, oder das Busy-Flag geprüft werden muß.

Siehe zur Problematik der LCD-Anzeigen auch die Arbeit von Harald Deichstetter an diesem Institut.

9.3 Fehlerhafte Funktion der SIMMs durch "power- und ground-bounce"

Bei dynamischem RAM werden die Informationen in unvorstellbar winzigen Kondensatoren gespeichert. Beim Lesen des Speicherinhaltes gehen die Informationen sofort verloren und werden daher während des Lesevorganges im Speicher-Chip in speziellen Latches zwischengespeichert und automatisch wieder neu gespeichert.

Das Auswerten der Ladungen der Kondensatoren ist ein äußerst kritischer Vorgang, der sehr empfindlich gegenüber Schwankungen der Versorgungsspannung ist.

Unglücklicherweise werden Schwankungen der Versorgungsspannungen unter anderem von eben diesen Lesevorgängen hervorgerufen, da diese kurzzeitig zu hoher Stromaufnahme führen. Dieses kurzzeitige Schwanken der Versorgungsschwankung wird als power- bzw. ground-bouncing bezeichnet. Vermutlich durch diesen Effekt funktionierten einige SIMMs nicht. Andere Exemplare waren dagegen unempfindlich und arbeiteten klaglos. Durch Einbau einiger Stützkondensatoren in die Spannungsversorgung konnte das Problem beseitigt werden.

10 Verwendete Hilfsmittel

10.1 Verwendete Hardware-Komponenten

- Host-PC mit serieller Schnittstelle. Betriebssystem DOS.
- Entwicklungs-PC. Betriebssystem Windows 95.
- Evaluation board EVA167 der Firma Ertec zum Steuern des SIMM-Testers.

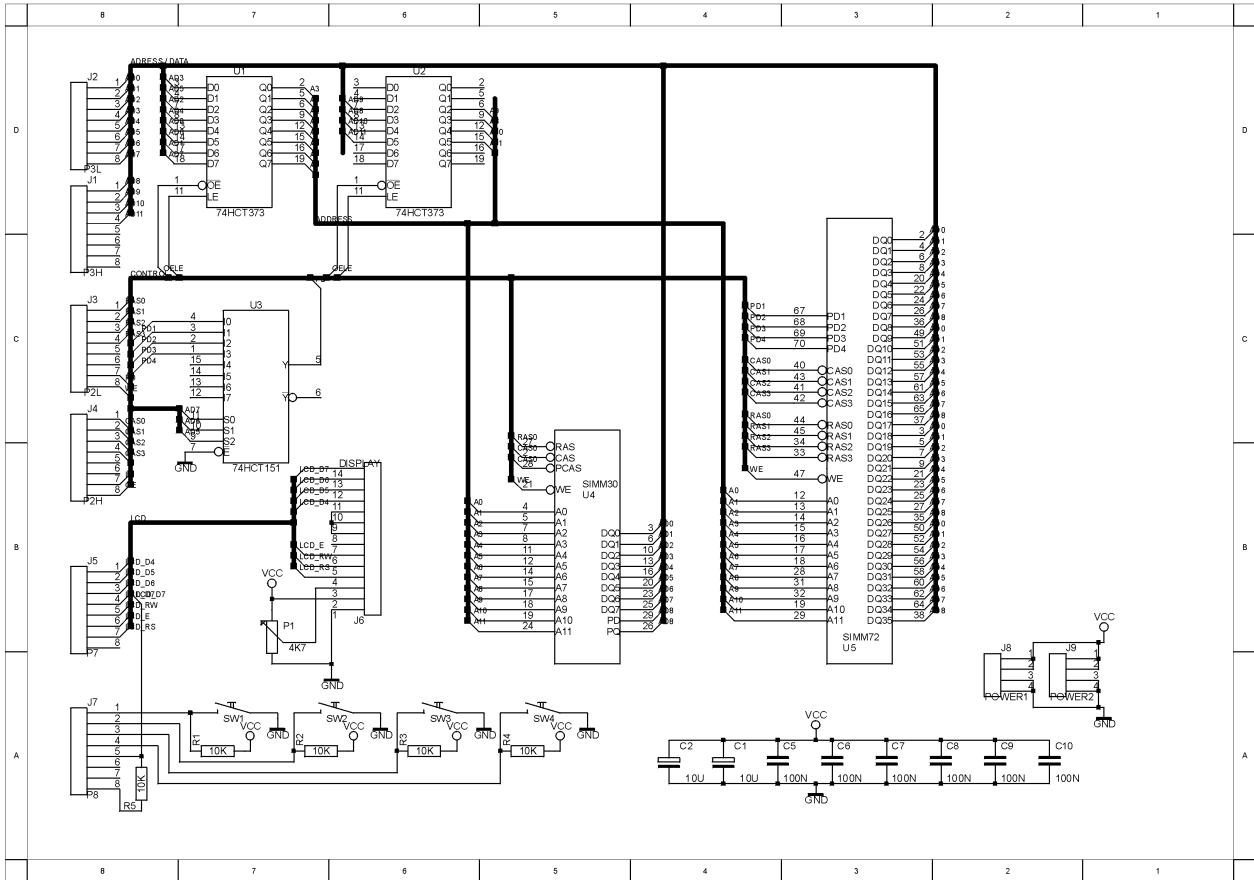
10.2 Verwendete Komponenten und Software-Tools

- Monitorsoftware MON16X der Firma Ertec zur Kommunikation mit dem Evaluation board.
- Digital Application Engineer DAVE Version 1.0 der Firma Siemens zur Konfiguration des Mikrocontrollers.
- Keil C166 Version 3.00 Entwicklungsumgebung zur Erstellung und Compilation der Programme für den Mikrocontroller.
- Ultiboard Entwicklungsumgebung für Schaltplan und Platinenlayout.

10.3 Verwendete Literatur

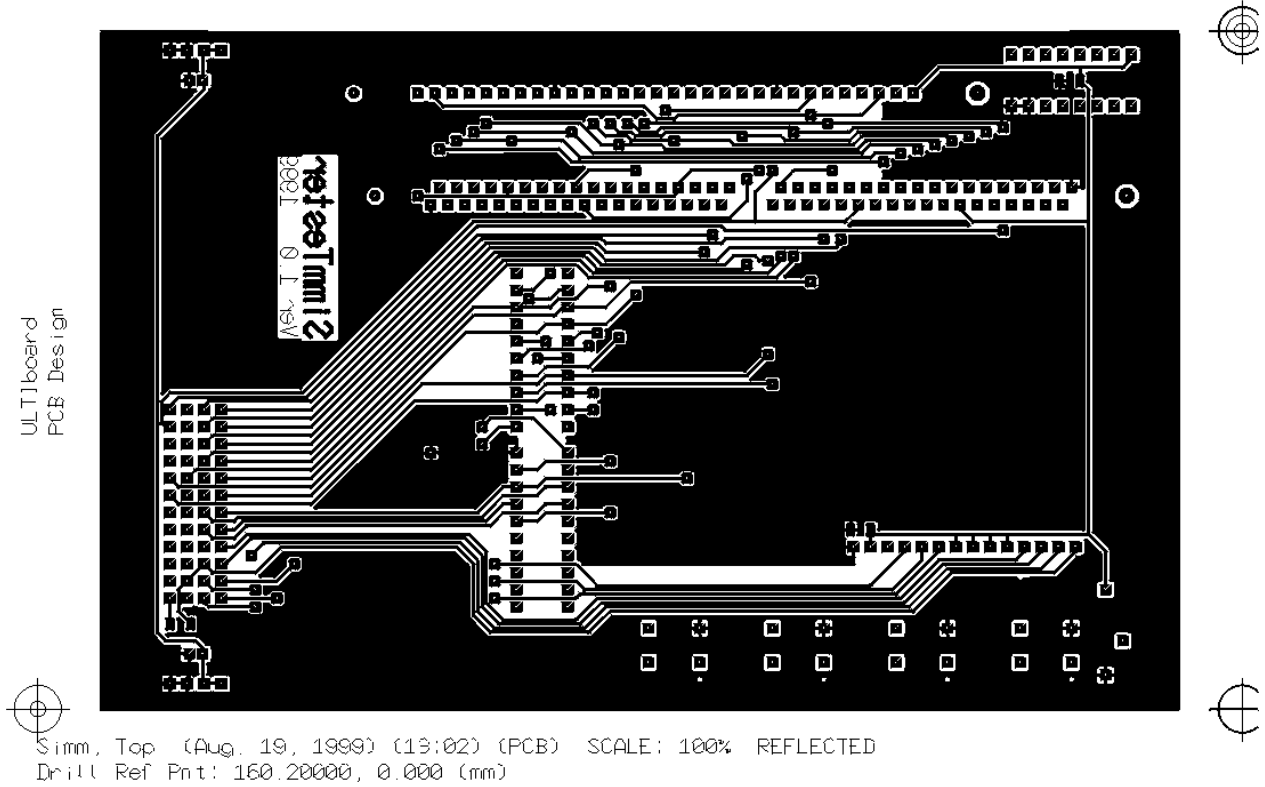
- Siemens C167 Derivatives 16_Bit CMOS Single-Chip Microcontrollers User's Manual; 03.96; Version 2.0
- Micron DRAM Data Book 1997
- Walter Waldner: Erfolgreich starten mit dem Siemens C167-Starterkit und dem Software-Entwicklungssystem von Keil; PCNEWS 60; November 1998
- SIMM-Tester; Elektor 2/96
- Memory Guide; http://www.pbdatacom.se/empire/hardware/memory/memory_guide.html
- 4M and smaller DRAM by Manufacturer and Part Number; http://www.ridgecrest.ca.us/~markee/Hardware_stuff/memory/SIZE/064m_manu.htm
- Samsung Electronics: KM41C4000D CMOS DRAM Data sheet
- Samsung Electronics: KMM5321200C2W DRAM Module Data sheet
- Support; <http://www.drix-be/numbers.htm>
- Memory Guide; <http://www.carleton.ca/~taraujo/raven/mem.html>

11 Anhang A: Schaltplan

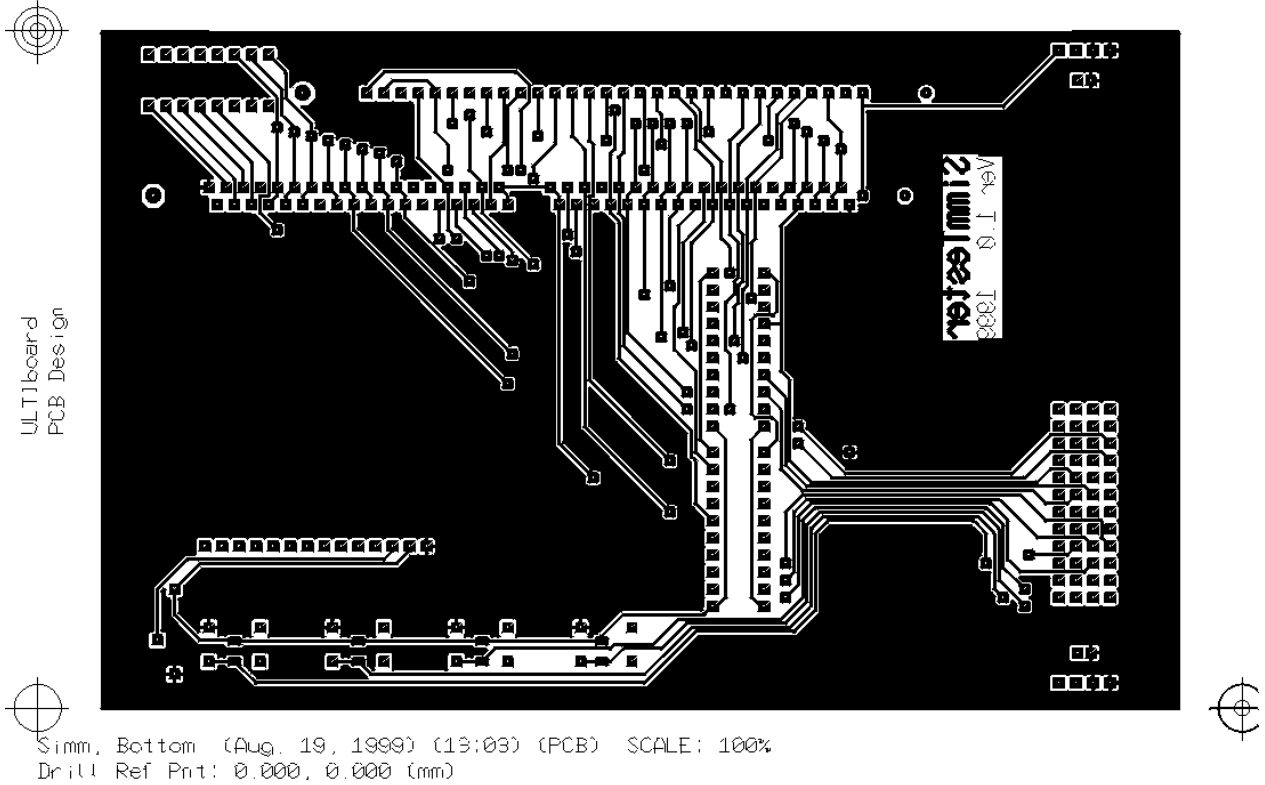


12 Anhang B: Platinen-Layout

Oberseite:

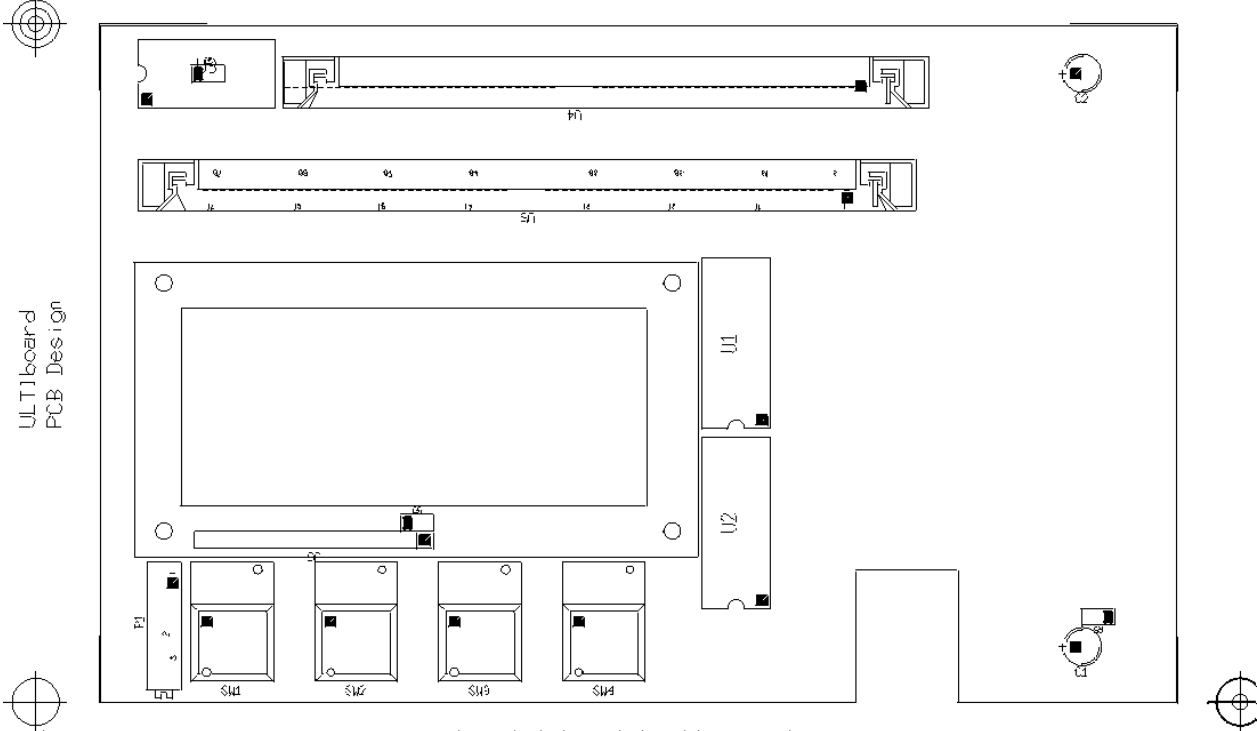


Unterseite:



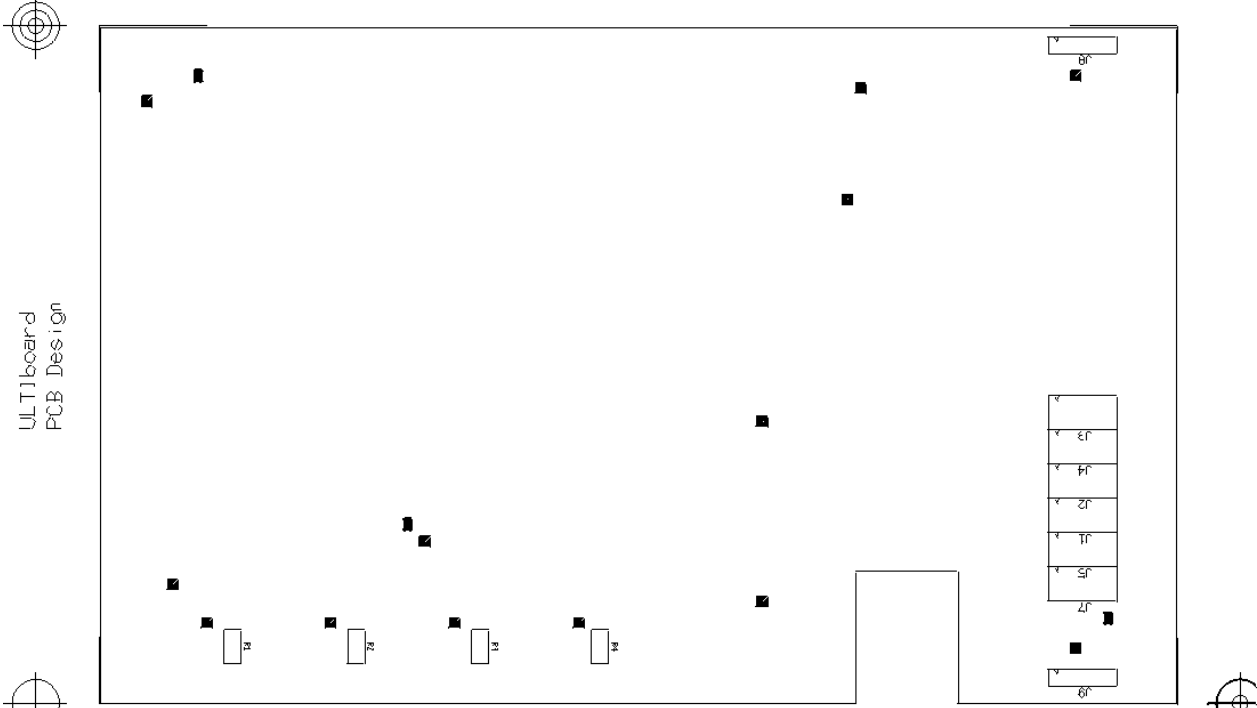
13 Anhang C: Bestückungsplan

Oberseite:



Simm, Silkscreen (Aug. 19, 1999) (13:00) (PCB) SCALE: 100%
 Drill Ref Pnt: 0.000, 0.000 (mm)

Unterseite:



Simm, Silkscreen (Aug. 19, 1999) (13:00) (PCB) SCALE: 100%
 Drill Ref Pnt: 0.000, 0.000 (mm)

14Anhang D: Stückliste

B I L L O F M A T E R I A L S

Project name: C:\SIMMTest\SimmmTest.UTSCH

Time & date: 12:49:23 27-09-1999

#	Device	Value	Package type	Refdes
1	74HCT151	74HCT151	DIP16	U3
2	74HCT373	74HCT373	DIP20	U1
				U2
6	CAPACITOR	100N	R/C1206_W	C5
				C6
				C7
				C8
				C9
				C10
1	DISPLAY	HDR_14	LCD14_16X2	J6
2	ELCO	10U	ELKO5R2	C2
				C1
1	P2H	P2B	HEADER2X4	J4
1	P2L	P2A	HEADER2X4	J3
1	P3H	P3B	HEADER2X4	J1
1	P3L	P3A	HEADER2X4	J2
1	P7	P7	HEADER2X4	J5
1	P8	P8	HEADER2X4	J7
1	POT.METER_1	4K7	SPITRI19X5	P1
1	POWER1	POWER1	HEADER1X4	J8
1	POWER2	POWER2	HEADER1X4	J9
5	RESISTOR	10K	R/C1206_W	R1
				R2
				R3
				R4
				R5
1	SIMM30	SIMM30	SIMM30VR50	U4
1	SIMM72	SIMM72	SIMM72VR50	U5
4	SWITCH	SW_SPST	DIGTAS1	SW1
				SW2
				SW3
				SW4

Current processed Variant : DEFAULT

B I L L O F M A T E R I A L S

Project name: C:\SIMMTest\SimmmTest.UTSCH

Time & date: 12:49:23 27-09-1999

#	Device	Value	Package type	Refdes
---	--------	-------	--------------	--------

15 Anhang E: Dateien auf der CD

SimmDocu.doc: Dieses Dokument

Simm.UTSCH: ULTIcap Schaltplan

Simm.DDF: ULTIboard Layout

Simm.bom: Stückliste

SimmTest.dav: DAvE-Datei

Docu.rtf: Dokumentation von DAvE generiert

Simmtest.prj: Projektdatei für die Keil Entwicklungsumgebung

Simmtest.h86: Hex-Datei zum Laden auf das Entwicklungsboard

Src: Verzeichnis mit allen Quellcodes und dem Makefile