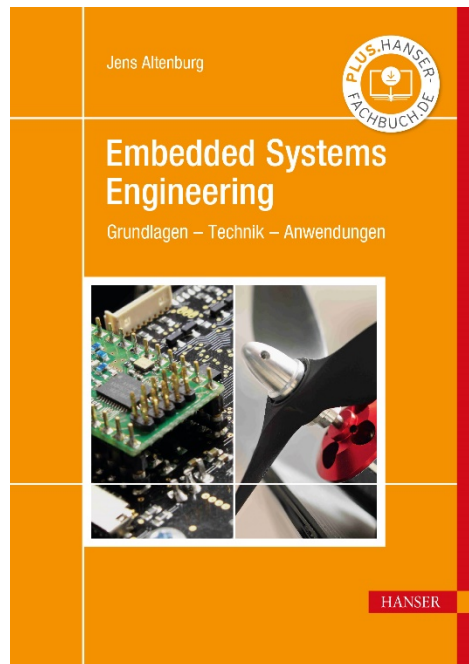


HANSER



Leseprobe

zu

Embedded Systems Engineering

von Jens Altenburg

Print-ISBN: 978-3-446-46735-4
E-Book-ISBN: 978-3-446-46842-9

Weitere Informationen und Bestellungen unter
<http://www.hanser-fachbuch.de/978-3-446-46735-4>

sowie im Buchhandel

© Carl Hanser Verlag, München

Inhalt

Danksagung	IX
Über den Autor	XI
1 Einführung	1
2 Schaltungstechnische Grundlagen	5
2.1 Passive Bauteile – R, L und C	5
2.2 Aktive Komponenten	9
2.3 Schaltungsberechnung und Simulation	12
3 Schaltungstechnik	21
3.1 Analoge Schaltungen	21
3.2 Der Transistor als Schalter	23
3.3 Digitale Grundfunktionen	24
3.3.1 Negation	26
3.3.2 UND-Verknüpfung	27
3.3.3 ODER-Verknüpfung	28
3.3.4 TTL-Bausteine als industrielle Lösung	29
4 Digitale Schaltungssynthese	35
4.1 Aufstellen der logischen Schaltfunktionen	35
4.2 Schaltungssynthese mittels kanonischer Normalform	37
4.3 Simulation digitaler Schaltungen mit Logisim	39
4.4 Boolesche Algebra	41
4.5 Minimierung nach Karnaugh-Veitch	42

4.6	Zahlensysteme und deren Darstellung	45
4.7	Codierung und Fehlerkorrektur	49
4.8	Schaltnetze	57
4.8.1	Codeumsetzer	57
4.8.2	Multiplexer und Demultiplexer	60
4.8.3	Arithmetische Grundfunktionen	65
4.9	Schaltwerke	71
4.9.1	Flip-Flops als Grundbausteine der Schaltwerke	74
4.9.2	Schieberegister	81
4.9.3	Asynchrönzähler	85
4.9.4	Synchrönzähler	87
4.9.5	Sonderfunktionen	91
5	Hardware eingebetteter Systeme	95
5.1	Mikrocontroller vs. Mikroprozessor	98
5.2	CPU	99
5.3	Speicherlayout und Speichertypen für MCUs	101
5.3.1	Programmspeicher (ROM)	104
5.3.2	Arbeitsspeicher (RAM)	109
5.4	Takterzeugung	111
5.5	Busankopplung und digitale IOs	113
5.6	Interrupt-Handling	124
5.7	Peripheriemodule	133
5.7.1	Zeitgeber und Zähler	133
5.7.2	Analog-Digital-Wandler	141
5.7.3	Serielle Schnittstellen	144
5.7.4	Sonderfunktionen	154
6	Programmierung eingebetteter Systeme	157
6.1	Notation und Programmierstil in C	157
6.2	Speicherlayout und Variablenhandling in Funktionen	164
6.3	Erste Schritte mit Mbed	168
6.4	Rowley Crossworks für Experten	173
6.5	Multitasking auf dem Mikrocontroller	179

7	Anwendungen eingebetteter Systeme	191
7.1	Vom Einplatinenrechner zum Mikrocontroller	191
7.2	„Blick zurück nach vorn“ – Tiny BASIC mit U883	194
7.3	Kryptografie in BASIC mit einer 8-Bit-MCU	198
7.4	ROBOTECH als Urahn der Roboter	206
7.5	Mit Robo Eye auf Pirsch	208
7.5.1	Hardwarebasis des Robotersystems	208
7.5.2	Das geheime Auge – „Spy on your nerd friends!“	214
7.5.3	Bildübertragung mit dem Robo Eye	218
7.5.4	Objektdetektion in Echtzeit auf einer 16-Bit-MCU	221
7.5.5	Bilddatenkompression mittels BTC	227
7.6	„Wir lassen Intelligenz fliegen“ – unbemannte Luftfahrzeuge stellen sich vor	232
7.6.1	„Trial und Error“ – des Ingenieurs schwarze Kunst	232
7.6.2	Lagestabilisierung für Dummies	235
7.6.3	Flugsimulation im Labor	242
7.6.4	Der ARM Cortex-M3 bekommt Flügel	247
8	PIRX^{AT} – vom Prototyp zum UAV-Testvehikel	257
8.1	Designkonzept eines UAV	257
8.2	Echtzeit-Multitaskingsystem nicht nur für Autopiloten	262
8.3	Das theoretische Modell als Grundlage der „Fliegekunst“	270
8.4	Testhardware für den Autopiloten PIRX ^{AT+}	275
8.5	Flugerprobung	280
9	Zusammenfassung und Ausblick	285
10	Übungsaufgaben mit Lösungsweg	289
	Literaturverzeichnis	311
	Index	317

Über den Autor

Nach dem Abitur und einem kurzen Ausflug in die Verteidigung des Vaterlandes, habe ich von 1985 bis 1990 Informationstechnik an der damaligen Technischen Hochschule Ilmenau studiert. Mit dem Abschluss als Diplomingenieur begann die berufliche Karriere im VEB Robotron Büromaschinenwerk Sömmerda als Entwicklungsingenieur. Frisch verheiratet, ein kleines Kind zu Hause und sich dann dem Niedergang eines Kombines entgegenzustemmen, das war eine herausfordernde Zeit.

Viel lieber hätte ich ein Sabbatical in Neuseeland absolviert, bedauerlicherweise wusste ich nicht, wo man das dafür notwendige Formular findet und wie es auszufüllen ist. Den Untergang von Robotron konnte ich leider nicht verhindern, wahrscheinlich hätte ich auch die Titanic nicht retten können. Zumindest gelang es mir, meinen damaligen Chef von meinen Fähigkeiten so zu überzeugen, dass ich die Chance erhielt, in der Ausgründung der Entwicklungsabteilung mittun zu dürfen.

Meine hochgeschätzte Frau behauptet, nach ein paar Jahren Tätigkeit in einer Firma sticht mich der Hafer, und sie macht dazu eine eindeutige Bewegung mit dem Zeigefinger an den Kopf. Tatsächlich habe ich nach fast zehn Jahren Hardwareentwicklung meinen Tätigkeitsschwerpunkt in die Software verlagert, und zwar in die Entwicklung eingebetteter Systeme für die Automobilindustrie. Trotz vieler reicher Erfahrungen habe ich es da nicht lange ausgehalten. Als Kodierknecht wollte ich nicht mein ganzes Berufsleben verbringen. Ein Gutes hatte die Zeit jedoch. Neben meiner beruflichen Tätigkeit habe ich an der TU Ilmenau promoviert. Auf die Bitte um Unterstützung durch den damaligen Arbeitgeber erhielt ich die Auskunft: „Was Sie in Ihrer Freizeit treiben, ist mir egal!“. Das war eindeutig, deswegen habe ich 2003 mein Ränzchen geschnürt und mein Glück in einem interessanteren Unternehmen versucht. Eigentlich sollte nur eine kleinere Idee in einer begrenzten Entwicklungszeit umgesetzt werden. Daraus sind dann ebenfalls wieder fast zehn Jahre Berufstätigkeit geworden. Ich denke immer noch mit großer Hochachtung an die damalige Geschäftsführung zurück, die sich mit so einem illustren Charakter wie dem meinen abgab.

Zur letzten Änderung meiner Karriere gab ein Kollege anlässlich eines Treffens auf einer Messe Anlass. Er wollte mir unbedingt seine neue Visitenkarte zeigen. „Gerald, wir kennen uns seit Jahren, was interessiert mich dein neuer Arbeitgeber?“ Als ich das Kärtchen dann doch in Empfang nahm, hat es mich beinahe umgehauen: „Prof. Dr.-Ing. – Wahnsinn! Wie hast du das denn gemacht?“ Zwei Jahre später stand ich selbst vor der Entscheidung, eine Professur nebenan oder eine andere ziemlich weit weg anzutreten. 2012 nahm ich die andere. Sie erinnern sich noch an die typische Bewegung meiner Frau, die mit dem Finger und dem Kopf? Fernpendeln, nachdem ich viele Jahre mit dem Fahrrad zur Arbeit fuhr; zudem auch noch ziemlich weit weg. Dennoch, den Übergang in die akademische Welt habe ich bislang noch nicht bereut, aber es fehlen ja noch zwei Jahre bis zum Ablauf der zehnjährigen Wechselfrist.

Von den packenden und schwierigen Aufgaben eines Ingenieurs handelt dieses Buch. Ich schrieb es in recht bewegten Zeiten. Zuerst als Handreichung zum Selbststudium für meine Studenten gedacht, hat sich das Ganze dann doch zu einem größeren Projekt gemausert. Innerhalb weniger Monate sind mehr als zwanzig Jahre Berufserfahrung und die eine oder andere Anekdote aus der Lehre destilliert und zu einem, ich hoffe, interessanten und lesenswerten Vademecum konzentriert worden.

1

Einführung

Die Idee zu diesem Lehrbuch entstand aus den Nachfragen vieler Studenten nach Begleitliteratur zu den Kursen „Grundlagen digitaler Schaltungstechnik (DIGI)“, „Mikroprozessortechnik (MPRO)“ sowie „hardwarenahe Programmierung (HAPO)“, die für den Erwerb des Bachelorabschlusses in Elektrotechnik an der TH Bingen absolviert werden müssen. Die vorhandene Fachliteratur [1, 2] wurde von den Studenten teilweise als inhaltlich zu umfangreich und gelegentlich auch als zu unspezifisch eingestuft, oder die Werke sind bereits vergriffen [3]. Obgleich ich diese Kritik nicht in allen Fällen nachvollziehen konnte, stellte sie doch einen Teil der Motivation zum Schreiben dieses Buches dar.

In den Grundlagenvorlesungen wird anfangs viel abstrakte Theorie vermittelt. Der Verweis auf die späteren Anwendungen soll die Studenten über das gelegentlich trockene Material „hinwegrösten“. Deswegen besteht das vorliegende Buch aus zwei eng miteinander verflochtenen Einheiten.

Zunächst werden die wichtigsten Grundlagen für das Verständnis digitaler Schaltungen und deren Anwendung im Kontext hardwarenaher Programmierung in „Eingebetteten Systemen“ dargelegt. Die Grundlagenkapitel (Kapitel 2 bis 5) beginnen mit zum Teil sehr populärwissenschaftlichen Erklärungen und starken Vereinfachungen. Doch lassen Sie sich nicht täuschen. Der Gradient der Steigerung der Komplexität der Inhalte könnte einer Exponentialfunktion ähneln.

In den späteren Kapiteln (Kapitel 6 bis 8) geht es um typische Anwendungen leistungsfähiger eingebetteter Systeme im Umfeld autonomer Miniaturroboter und unbemannter Luftfahrzeuge (unmanned aerial vehicles, UAV). Damit rechtfertigt sich auch der Titel des Buches. Es empfiehlt sich deswegen auch für die Lehre des Moduls „Embedded Systems (EMSY)“ im Masterstudiengang der TH Bingen und in vergleichbaren Studiengängen an anderen Hochschulen.

Dieser zweite Teil des Buches bildet beinahe eine Art Fortsetzung meiner Darlegungen aus dem Buch *Mobile Roboter* [4], welches vor mehr als 20 Jahren erstmals publiziert wurde. Es erscheint nahezu unglaublich, in welchem Maße die Steigerung der Leistungsfähigkeit der Mikrocontrollertechnik sowie der Sensortechnologie in den inzwischen vergangenen Jahren vorangeschritten ist. Hin und

wieder nehme ich mir deswegen die Freiheit, technikhistorische Reminiszenzen in die fachlichen Erläuterungen einfließen zu lassen. Nicht alle diese Bezüge verstehen sich als trockene technische Prosa, ganz im Gegenteil, mit diesen persönlichen Anmerkungen soll der Inhalt des Buches aufgelockert und auch für Fachlaien interessant und lesbar werden.

Diese teils nostalgischen Rückblicke sind dennoch nicht nur eine Art verklärte Vision, sondern sie dienen vorrangig der Sensibilisierung der jetzigen Schüler, Studenten und aller anderen Enthusiasten hinsichtlich des Ursprungs und der möglichen zukünftigen technischen Prozesse. Denn nur wenn wir untersuchen und verstehen, woher Dinge stammen, werden wir erkennen, wie sie sich entwickeln könnten, und nur daraus lassen sich einigermaßen belastbare Voraussagen für die Zukunft treffen.

Diese Rückblicke sind ausdrücklich nicht als Warnungen vor einer vermeintlichen Übertechnisierung zu verstehen. Ich persönlich vertraue vollständig der Ratio kommender Ingenieure als wichtige Gestalter einer auch in Zukunft leistungsfähigen Industrie sowie deren verantwortungsvollen Nutzung aller natürlich vorkommenden Ressourcen. Aus langer Berufserfahrung kann ich nur immer wieder bekräftigen: Am Ende gewinnt stets die Realität.

Alle geschilderten fachlichen Inhalte basieren auf sorgfältiger Recherche, eigenen Erfahrungen und Überprüfungen durch Nachrechnung bzw. Simulation. Sollten dennoch Fehler enthalten sein, gehen diese alle auf meine Kappe, und ich bin für entsprechende Hinweise dankbar.

Über die grundsätzliche fachliche Beschreibung hinaus sollen viele Beispiele das Gebiet der eingebetteten autonomen Systeme illustrieren. Vielfach sind Stromlaufpläne und Programmteile sehr detailliert angegeben. Diese Angaben sind alle erprobt und sollten auch erfahrenen Technikern und Ingenieuren den einen oder anderen interessanten Hinweis liefern.



Sämtliche Quellcodes und Simulationsbeispiele aus dem Buch stehen unter plus.hanser-fachbuch.de in ungekürzter Form bereit und lassen sich mit kostenlos im Internet verfügbaren Werkzeugen nutzen.

Darüber hinaus finden Sie einige der Programmbeispiele zum Thema MPRO als sofort nutzbare Projekte auf der Mbed-Webseite (os.mbed.com). Über das Suchfeld der Webseite können Sie nach den Stichwörtern „Hanser“ und „Altenburg“ suchen. Die Programme werden Ihnen dann zusammen mit einer kurzen Beschreibung angezeigt und können mit dem IMPORT-Button sofort in den eigenen Workspace übernommen werden.

In Kapitel 8 und 9 beziehe ich mich ausdrücklich auf Zuarbeiten früherer Studenten und späterer Mitarbeiter meiner Forschungsgruppe UAV an der Technischen Hochschule Bingen. Explizit seien hier Christopher Hilgert, Sarah Armbrüster,

Johannes von Eichel-Streiber, Christoph Weber und Andreas Gross (in chronologischer Abfolge) genannt.

Der Einsatz und die Anwendung der besprochenen Beispiele erstrecken sich über einen Zeitraum von circa 30 Jahren. Doch selbst bei kritischer Sicht auf die Historie sollte offenbar werden, dass sich zwar die eine oder andere Äußerlichkeit geändert hat, die wirklich relevanten Interna eingebetteter Systeme sind indes fast unverändert geblieben. Lediglich die Leistungsfähigkeit heutiger microcontroller units (MCUs) ist praktisch unbeschreiblich.

Ein kleines Detail am Rande: Mit Ausnahme von drei Abbildungen sind alle Fotos eigene Aufnahmen. Mir lag jedoch keine Abbildung eines Ringkernspeichers und kein rechtfreies Foto der Verschlüsselungsmaschine ENIGMA vor. Außerdem wollte ich gern eine Aufnahme des Mars-Rovers Sojourner im Buch unterbringen. Im Netz wurde ich fündig und so haben Jussi Kilpeläinen (Finnland), Simon Singh (Großbritannien) und Bert Ulrich (USA) dieses Buch durch die Bereitstellung von Abbildungen unterstützt. Das ist wirkliche kollegiale internationale Zusammenarbeit. Dafür meinen herzlichsten Dank.

In diesem Sinne wünsche ich allen pfiffigen Studenten und Schülern beim Selbststudium der Grundlagen moderner Digitaltechnik und beim Erwerb von Kenntnissen über Mikrocontroller viel Erfolg sowie allen anderen Lesern gutes Gelingen bei der Entwicklung eingebetteter Systeme.

Jens Altenburg, im August 2020

■ 3.3 Digitale Grundfunktionen

Am Anfang des globalen Informationszeitalters bestand die weltweite Kommunikation aus Morsezeichen. Kommunikation ist ein wichtiges Grundbedürfnis der modernen Zivilisationen. Ursprünglich mehrheitlich für militärische Zwecke entwickelt und eingesetzt, stand spätestens mit dem Beginn der industriellen Revolution der Informationsaustausch an der Spitze der Agenda. Gelegentlich wird sogar darüber diskutiert, ob nicht die großräumige Informationsverteilung eher die Ursache und nicht die Folge der industriellen Revolution sein könnte.

Zu Beginn stand die Frage der technologischen Basis der Datenkommunikation. Anschaulich wird dies in der Verlegung des ersten interkontinentalen Telegrafenkabels zwischen Europa und Nordamerika 1858 illustriert (siehe dazu [16]). Seinerzeit beinahe als ein Weltwunder angesehen, beschleunigte sich doch damit der Austausch schriftlicher Dokumente von bis dahin einigen Wochen Seereise auf eine praktisch sofortige Kommunikation. Mit diesen technologischen Großtaten erreichte die Elektrotechnik den ersten Gipfel ihres Leistungsvermögens.

Um die Wende zum 20. Jahrhundert gelang auch die Überbrückung großer Distanzen mittels drahtloser Telegrafie. Mit großen Wellenlängen, riesigen Antennen und geradezu gewaltigen Leistungen konnte man nun durch „Großfunkstellen“ weltweit kommunizieren. Das Militär, insbesondere die Marine, hatte selbstverständlich von Anfang an sehr großes Interesse an dieser neuzeitlichen Technik. Nur mithilfe der Funktelegrafie war die Koordinierung von Hochseeschiffen möglich.

Doch immer das Militär als Technologietreiber vorzuschieben, ist eine einseitige Sichtweise. Wie bereits erwähnt, ist die Wirtschaft ein großer Gewinner schneller Kommunikation. Vergleichsweise preiswerter Datenaustausch stärkt die Wirtschaftskraft und verleiht Wettbewerbsvorteile gegenüber der Konkurrenz ohne diese leistungsfähige Infrastruktur. Nicht zuletzt erlaubt die drahtlose Telegrafie auch den Austausch von Wetterdaten. Die international üblichen Wetter(kurz)symbole stammen aus jener Zeit.

Am Rande soll aber auch auf die mögliche Verwundbarkeit moderner Infrastrukturen hingewiesen werden. Neben menschlichem Versagen, z. B. Blackout der Energieversorgung durch verfehlte strategische Planung, sind immer noch natürliche Ereignisse mit potenziell katastrophalen Folgen, zumindest gedanklich, zu berücksichtigen. Wenn das derzeitige weltweite Kommunikationsnetz mit einem ähnlichen Phänomen wie dem „Carrington-Ereignis“ [17] konfrontiert würde, wären die negativen Folgen kaum abschätzbar.

Mit der Erfindung der Elektronenröhre gelang der Massendurchbruch im Kommunikationsbereich. Endlich war nicht nur die digitale Informationsübertragung in Form der Morsetelegrafie möglich, sondern der direkte Sprechkontakt zwischen

Menschen überall auf der Erde wurde Realität. Die große Zeit der analogen Nachrichtenübertragung hatte begonnen.



Historische Anmerkung 2

Mein erster Detektorempfänger ist leider längst im Schrott gelandet. Glücklicherweise fand sich an der TH Bingen noch ein Originalgerät mit Detektorkristall (Bild 3.3 Mitte) zusammen mit einem Kopfhörer, also Hightech aus den 1920ern.

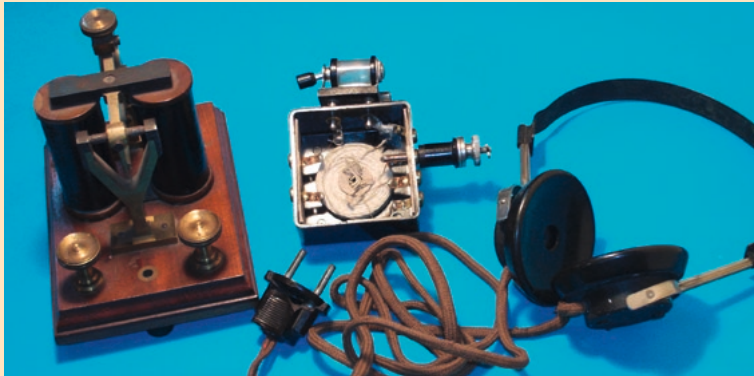


Bild 3.3 Historische Messinstrumente und Gerätschaften

Das links in Bild 3.3 befindliche Bauteil ist ein Telegrafenklopfer, der Vorläufer des Kopfhörers. Dieses Teil habe ich in einem Antiquitätengeschäft in Verona entdeckt. Der Verkäufer erzählte mir Wunderdinge über dieses angeblich aus den 1940ern stammende „Relais“. Ein bisschen haben wir über den Preis verhandelt, dann habe ich das Teil gekauft, ohne den Händler über seinen Irrtum aufzuklären. Der Klopfer ist deutlich älter als angegeben und für Sammler definitiv mehr wert als der gezahlte Kaufpreis. Damit ist ein singuläres Ereignis eingetreten: Ein Ingenieur führt einen ausgebufften (aber sehr sympathischen) Verkäufer hinter die Fichte! In dieser historischen Einmaligkeit ist dies praktisch vergleichbar mit dem Urknall in der Astronomie.

Mit der Einführung des Rundfunks, also dem Empfang drahtloser Sendungen für jedermann, wurde die Informationstechnik endgültig zum Massenphänomen. Digital war vorgestern – das neue Zeitalter war das Jahrhundert der Elektronenröhre, dessen war man sich bis in die 1950er-Jahre sicher. Das Aufkommen der Halbleitertechnik wurde vielerorts nicht als Konkurrenz zur Röhre wahrgenommen. Lediglich Spezialanwendungen wie Taschenradios für Halbstarke wurde ein Markt zugemessen.

Von diesen Prognosen völlig unbeeindruckt, forschten einige Ingenieure, von einigen der damaligen Stimmen eher als bessere „Bastler“ bezeichnet, an den Einsatz-

möglichkeiten der Festkörperphysik der Halbleiter. Diese Bastelfritzen (einige davon spätere Nobelpreisträger) präsentierten dann in den frühen 1960ern die ersten Serien funktionsfähiger „integrierter Schaltkreise“.

Auf einem Stück Germanium, später Silizium, wurden einige wenige Transistoren kombiniert; dass daraus in Zukunft Milliarden werden sollten, ahnte niemand. Im Gegensatz zur Elektronenröhre ist das Übertragungsverhalten eines Transistors deutlich nichtlinearer als die quadratische Kennlinie einer Triode. Diese zunächst scheinbar unwichtige und in der Verwendung in Verstärkern sogar nachteilige Eigenschaft wird beim Einsatz in der Digitaltechnik zum fundamentalen Vorteil.

Digitalrechner sind keine Erfindung der Gegenwart, die ersten brauchbaren Rechenmaschinen arbeiten mit mechanischen Zählelementen im digitalen Zehner-Zahlensystem [18]. Mit dem Aufkommen der Elektrotechnik fanden Relais und Dioden, später auch Röhren ihren Weg in den Digitalrechner.

Der Einsatz der Elektronenröhre in logischen Grundsaltungen blieb jedoch problematisch. Zwar half die Entwicklung spezieller Schaltröhren, die Eigenschaften der Logikbausteine zu verbessern. Das Problem der schlechten Schalteigenschaften, eine Elektronenröhre ist eben einfach besser für lineare Übertragungen geeignet als für extrem nichtlineare Sprungvorgänge¹, verblieb jedoch weiterhin.

Es klingt beinahe wie eine Ironie der Technikgeschichte, dass die anfangs eher belächelten Transistoren zügig für Logikeinheiten eingesetzt wurden. Die relativ großen Volumina von Röhrenrechnern störten eher weniger², die erhebliche Heizleistung war in Zeiten vor der Klimakrise unwichtig, aber die Schalteigenschaften und später die Zuverlässigkeit der Halbleiter wurden zu unschlagbaren Argumenten.

3.3.1 Negation

Die einfachste logische Funktion ist die Invertierung des logischen Eingangszustands. Aus logisch „0“ wird logisch „1“ und umgekehrt. Wie die Logikpotenziale im konkreten Fall technisch realisiert werden, ist zunächst unwichtig. Im Folgen-

¹ Die (fast ideal) quadratische Übertragungskennlinie von Elektronenröhren ist deswegen einer der Hauptgründe für hochqualitative Leistungsverstärker im Audibereich. Richtige „Fledermausohren“ können wahrscheinlich tatsächlich den „warmen“ Röhrenklang vom „kalten“ Sound des Transistorverstärkers unterscheiden. Ich persönlich tendiere eher zu großen eisenhaltigen Netztransformatoren und MOSFETs in der Endstufe. Ein guter Tipp: Hochwertige Endstufen für den verwöhnten Audiophilen kauft man nach Gewicht – je schwerer, desto besser! Eine gute Lautsprecherbox sollte nicht unter 500 € pro Stück (Stand 2020) kosten. Alternativ versucht man, Monitorboxen des Typs BR25 von RFT Geithein zu ergattern. Diese sind aber auch mehr als 30 Jahre nach Produktionseinstellung nicht billiger. Das Geld für sauerstofffreie Kupferlitze ist, genauso wie für Atomstromfilter in der Netzzuleitung, zum Fenster hinausgeworfen.

² Solange der Röhrenrechner ENIAC bei einer Aufstellfläche von 10 × 17 Metern 27 Tonnen wiegt, kommt es auf 50 cm³ und 100 g pro Flip-Flop wirklich nicht an.

den wird dann von logisch „1“ gesprochen, wenn eine definierte Spannung anliegt bzw. bei logisch „0“ kein Potenzial messbar ist.

Die technische Umsetzung zeigt Bild 3.4. Der Transistor arbeitet in der Sättigung. Ein Basisstrom im Eingangskreis führt zum vollständigen Durchsteuern der Kollektor-Emitter-Strecke. Der Basisstrom als Folge der logischen „1“ am Eingang resultiert in einer logischen „0“ am Ausgang. Die Schaltung fungiert damit als Inverter.

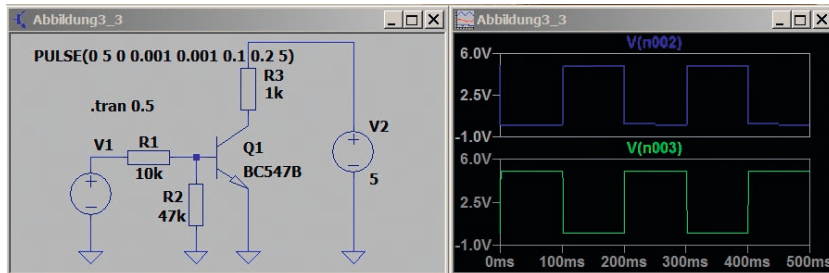


Bild 3.4 Inverter in Emitterschaltung: V(n002) und V(n003) zeigen das invertierende Verhalten der Logikschaltung. Der Transistor Q1 arbeitet als Schalter in der Sättigung.

Noch eine Ergänzung zu den Logiksignalen: In der Technik wird die „1“ häufig als „HIGH“ und die „0“ als „LOW“ bezeichnet. Dahinter verbergen sich je nach Herstellungstechnologie konkrete Spannungslevel. Weitere Informationen hierzu finden Sie in Abschnitt 3.3.4.

3.3.2 UND-Verknüpfung

In vielen technischen Anwendungen müssen mehrere Eingangssignale miteinander verknüpft werden. Als einfaches Beispiel soll die Bedienung einer Blechpresse für Karosserieteile genannt werden. In diesem Fall wird eine hydraulische Presse mit einem Blechrohling bestückt und dann in die gewünschte Form gebracht. In [19] ist ein Beispiel dafür zu sehen. Die Bedienung dieser Anlage erfolgt manuell. Beim Start des Pressvorgangs muss der Arbeitsraum frei sein. Ab Minute 1:16 im Video kann man erkennen, dass deswegen der Bediener zwei Tasten betätigen muss. Erst wenn Taste 1 UND Taste 2 betätigt werden, fährt der Pressstempel nach unten.

Elektrisch ist dies zunächst recht einfach durch eine Reihenschaltung beider Tasten realisierbar. In der schaltungstechnischen Umsetzung wird es etwas komplizierter. Laut Definition ergeben „1“ UND „1“ am Eingang eine „1“ am Ausgang.

In Bild 3.5 ist eine sogenannte Wired-AND-Schaltung dargestellt. Die beiden Eingangssignale werden mit den Spannungsquellen V1 und V2 generiert. Zwecks möglichst einfacher Darstellung erzeugen die Spannungsquellen ein phasenverschobenes Rechtecksignal. Dann, und nur dann, wenn beide Eingänge ein HIGH-Level aufweisen, ist auch am Ausgang ein HIGH-Potenzial zu sehen. Die beiden Dioden verknüpfen demnach V1 UND V3 zum gewünschten Logiksignal. Damit entsteht die benötigte Verknüpfung als Verschaltung von Halbleiterbauelementen.

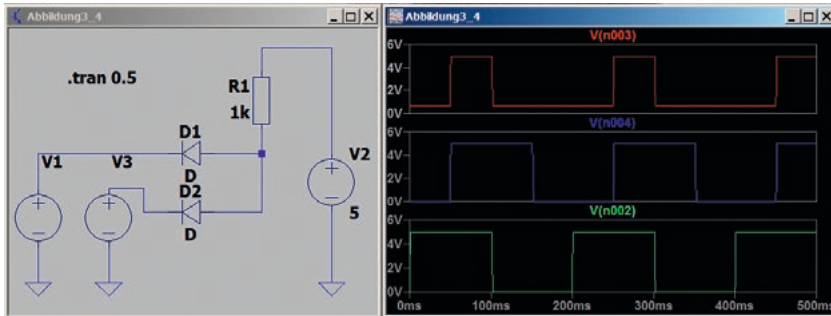


Bild 3.5 UND-Verknüpfung mit zwei Dioden und Spannungsverlauf am Eingang (V1, V3) und am Summationspunkt D1, D2 und R1

3.3.3 ODER-Verknüpfung

Als weitere logische Grundschaltung fungiert die ODER-Verknüpfung. Als klassisches Beispiel hierfür sei die Treppenhausbeleuchtung in einem mehrstöckigen Gebäude genannt. Von jeder Etage aus soll das Auslösen des Treppenlichtautomaten erfolgen können.

Der Unterschied der Spannungsverläufe zu Bild 3.6 ist offensichtlich. Das Ausgangssignal V(n002) ist nur dann logisch „0“ (LOW-Level), wenn beide Eingangssignale (V1/V2) ebenfalls „0“ sind. Solange V1 ODER V2 logisch „1“ sind, ist auch der Ausgang logisch „1“.

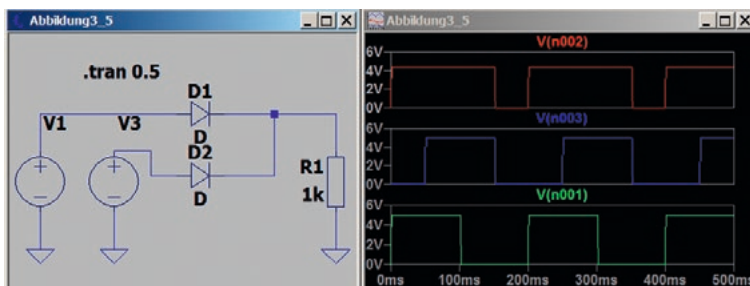


Bild 3.6 ODER-Verknüpfung mit zwei Dioden und Spannungsverlauf am Eingang (V1, V3) und am Summationspunkt D1, D2 und R1

Auf das Beispiel Treppenlichtsteuerung bezogen, wird klar, dass von jeder Etage aus das Startsignal für die Lichtsteuerung erzeugt werden kann.

Im Übrigen müssen die geschilderten logischen Verknüpfungen nicht zwangsläufig aus zwei Eingängen bestehen. Sehr leicht kann durch Hinzufügen von weiteren Dioden die Anzahl der Eingänge (nahezu) beliebig erweitert werden. In allen Fällen wirken die Eingangssignale auf den gemeinsamen Ausgang.

3.3.4 TTL-Bausteine als industrielle Lösung

In den genannten Beispielen sind die schaltungstechnischen Effekte der einfachen Diodennetzwerke stark vereinfacht worden. Eine Weiterverarbeitung der Signale war zunächst nicht vorgesehen. Aus der Simulation war aber schon nach der ersten Stufe eine leichte Verfälschung der Spannungspegel ersichtlich.

Von den angenommenen Idealwerten der Spannungen für HIGH = 5 V und LOW = 0 V kann bereits nach der ersten UND-Verknüpfung nicht mehr gesprochen werden. Der Blick in die Spannungslage von V(n003) in Bild 3.5 zeigt die Erhöhung des LOW-Levels um die Flussspannung der Eingangsdiode. Ähnliches zeigt sich in Bild 3.6. Hier verringert sich die HIGH-Spannung im Ausgang der ODER-Verknüpfung ebenfalls um die Flussspannung der Diode.

Da in der technischen Anwendung komplizierter Logikschaltungen mit einer Kaskade logischer Verknüpfungen zu rechnen ist, müssen die Signalpegel nach dem Durchlaufen der entsprechenden Verknüpfungen wieder regeneriert werden.

Mit dem Inverter existiert bereits eine Schaltung, die als „Digitalverstärker“ verwendet werden kann. Entsprechend der Dimensionierung des Basisvorwiderstands kann das Maß der Sättigung des Schalttransistors in weiten Grenzen eingestellt werden. Eine Variation des Basisstroms innerhalb dieser Grenzwerte hat keine Spannungsänderung am Ausgang des Inverters zur Folge. Der Inverter bzw. die Negation im logischen Sinne ist deswegen ein idealer Signalrestaurator. Einen nachteiligen Aspekt hat dieses Verfahren. Infolge der Sättigung mit Ladungsträgern zwischen den Grenzschichten im Transistor verlängert sich die Schaltzeit gegenüber einer Verstärkerstufe ohne Sättigung. Das zweite Resultat, die zusätzliche Negation der logischen Verknüpfung, ist nur auf den ersten Blick nachteilig.

Die Erweiterung der UND-Funktion mit dem Inverter zeigt das Schaltbild in Bild 3.7. Die Ausgangsspannung V(n003) in Bild 3.5 ist um den Betrag der Flussspannung der Schaltdioden angestiegen. Die Ausgangsspannung V(n002) in Bild 3.7 weist diesen Spannungsoffset nicht mehr auf. Lediglich die Spannungswerte sind zueinander invertiert. Aus dem UND wird ein Nicht-UND. Im internationalen Sprachgebrauch wird anstelle des Inverters von einem Negator gesprochen. Das ODER wird zu OR und das UND zu AND. Die Kombination UND + Negation wird jetzt als NAND bezeichnet.

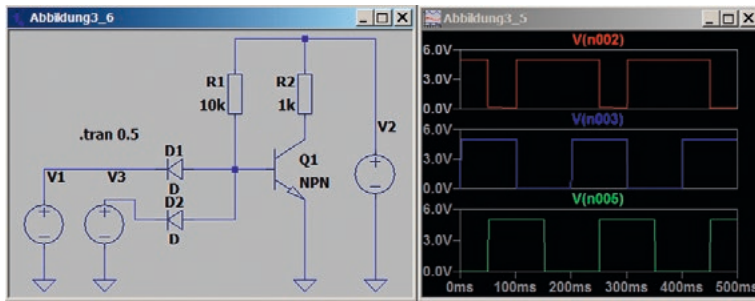


Bild 3.7 UND-Verknüpfung aus zwei Dioden und nachfolgender Negation zur Signalkonditionierung

Der Hauptunterschied zwischen analoger und digitaler Schaltungstechnik liegt in der Reduktion des analogen Kontinuums aller möglichen Spannungswerte in einem definierten Bereich und damit in einer diskreten Abbildung der Signale mit einer definierten Anzahl von Werten.

Da alle praktisch realisierbaren Systeme toleranzbehaftet sind, ist es für die technische Umsetzung von digitalen Schaltungen erforderlich, die Grenzbereiche der Erkennung von logisch HIGH bzw. LOW klar zu definieren.

Im Verlaufe der Entwicklung integrierter Bausteine ist eine Reihe von Technologiestufen durchlaufen worden. Es führt zu weit, an dieser Stelle ins Detail zu gehen, als wesentlichste Schritte sind vom schaltungstechnischen Ansatz her die RTL- bzw. DTL- und die TTL-Technologie zu werten. Des Weiteren spricht man von unipolaren bzw. bipolaren integrierten Schaltkreisen (integrated circuit, IC).

Die Abkürzung RTL steht für resistor transistor logic, das bedeutet, zur Realisierung eines derartigen IC sind Widerstände und Transistoren auf dem Silizium zu integrieren. Die ersten in Stückzahlen produzierten und für zivile Zwecke eingesetzten Bausteine waren Dreifach-NOR-Gatter der Firma Fairchild für das Apollo-Mondlandeprojekt. Pro Schaltkreis waren jeweils zwei Dreifach-NOR-Gatter integriert. Ein einzelner Baustein kostete nach [20] etwa 150 \$ nach heutiger Kaufkraft. Nur staatlich geförderte Prestigeprojekte, z.B. der Apollo Guidance Computer, konnten sich solche Kosten leisten. Bei geschätzten 2800 IC [21] wären dies ca. 420 000 \$ pro Apollo-Computer³.

Aus dem Stromlaufplan werden einige Ähnlichkeiten zu den bereits angegebenen Logikbaustufen ersichtlich (Bild 3.8). Anstelle der Diodenverknüpfung der ODER-Funktion wird jedoch gleich auf die Transistorstufen gegangen. Damit entfällt die

³ Zum Vergleich: Ein 32-Bit-ARM Cortex-M0, die leistungsschwächste ARM-CPU, benötigt etwa 12.000 Gatter-Äquivalente, also rund 6000 Dreifach-NOR. Das wären glatt 900 000 \$ für eine einzige CPU. Eine CPU der STM32F0-Serie ist in Einzelstücken für ca. 1,50 € erhältlich. Auf Serienstückzahlen hochgerechnet fällt der Preis deutlich unter einen Euro. Mit anderen Worten, eine CPU, vergleichbar der des Apolloprojekts, kostet etwa ein Millionstel dessen, was 1966 zu berappen war. Das nenne ich einmal Fortschritt!

Notwendigkeit, drei unterschiedliche Bauteiltypen zu integrieren. In [23] und [24] werden dazu einige ergänzende Informationen gegeben, in [23] erhält man einen Blick auf den Chip des NOR-Gatters und in [24], einem kurzen Videoclip, erfährt man interessante Zusammenhänge zur Funktion und zum Aufbau des Bausteins. Für tiefere Einblicke in den Apollo-Computer selbst sei auf [25] verwiesen.

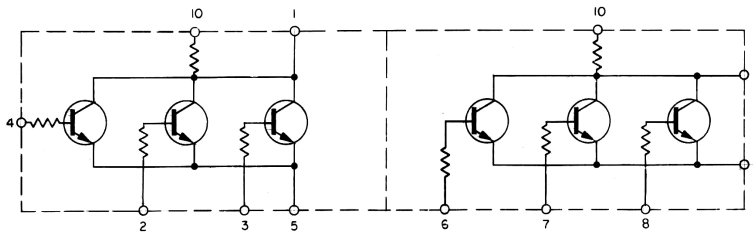


Bild 3.8 Stromlaufplan des 2 × Dreifach-NOR-IC für den Apollo Guidance Computer; Quelle: [22]

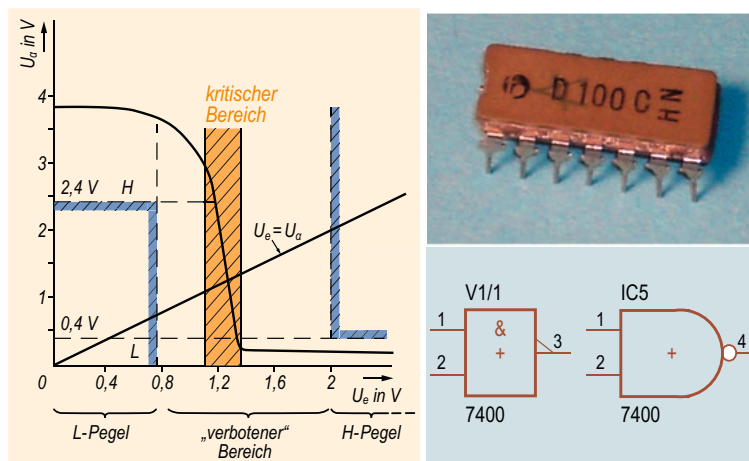
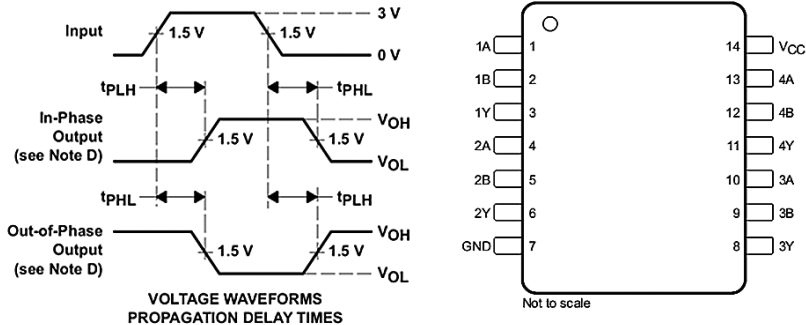


Bild 3.9 Übertragungskennlinie eines NAND-Gatters, Package und Logiksymbole; angelehnt an [26]

Mit dem Hinzufügen von Dioden erweitert sich die RTL- zur DTL-Logik. Einige Nachteile, wie z. B. die relativ hohe Verlustleistung der RTL-Bausteine, wurden ausgemerzt. Gleichzeitig stieg die Verarbeitungsgeschwindigkeit an, d. h., die Durchlaufverzögerung wurde kleiner (Bild 3.10).



$V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$, and over operating free-air temperature range (unless otherwise noted). See Figure 2.

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	A or B	Y	$R_L = 400\ \Omega$ and $C_L = 15\ \text{pF}$		11	22	ns
t_{PHL}					7	15	

Bild 3.10 Zeitverlauf zwischen Eingangs- und Ausgangspegel an einem TTL-Gatter: Rechts im Bild ist das Package des Bausteins mit der Zuordnung der Pins zu den Gattern zu sehen.

Der Ersatz der Dioden durch einen Multi-Emitter-Transistor führte zur TTL-Logik (transistor-transistor logic) und wurde zum Durchbruch in der Anwendung digitaler integrierter Schaltungen. Als Betriebsspannung wurde 5 V festgelegt. Die Übertragungskennlinie eines TTL-NAND-Gatters zeigt Bild 3.9.

In Bild 3.10 sind ein TTL-Baustein⁴ sowie die international üblichen Schaltzeichen zu sehen. Das linke Symbol entspricht der DIN, rechts ist das US-amerikanische Symbol angegeben.⁵

Spannungen zwischen 0 und 0.8 V werden als LOW-Signal und Pegel ab 2.4 V als HIGH-Signal erkannt. Damit sind technisch akzeptable Toleranzen vorgegeben. Der Spannungsbereich zwischen 0.8 und 2.0 V wird als „Verboten“ gekennzeichnet. Sämtliche Übergänge zwischen HIGH und LOW bzw. umgekehrt müssen diesen Bereich so schnell wie möglich durchlaufen. Dies ist durch geeignete Schaltungsmaßnahmen sicherzustellen. Für die Praxis bedeutet dies, dass in TTL-Schaltungen ausschließlich mit diskreten Pegeln und schnellen Impulsübergängen gearbeitet wird.

⁴ Für ein neuzeitliches Fachbuch gehört es zum guten Stil, auf originales Material zurückzugreifen. Für die Abbildung eines TTL-Schaltkreises stand leider kein Baustein der Firma Texas Instruments aus der Zeit der Serieneinführung (1964) zur Verfügung. Es wurde deswegen auf ein sorgfältig kopiertes Modell aus einheimischer Produktion zurückgegriffen. Der abgebildete D100C aus dem VEB Halbleiterwerk Frankfurt Oder (ehemals DDR) ist praktisch fabrikenü und wurde von mir persönlich während eines Industriepraktikums 1988 aus einer Schachtel für zukünftige Anwendungen entnommen, z. B. als Fotomodell für ein Fachbuch. Laut [72] entspricht das codierte Herstellungsdatum „HN“ dem November 1976. Also ist dies ein klarer Fall für die wohlverdiente Pension im nächsten Jahr.

⁵ Beide Symbole werden in einer kunterbunten Mischung in den unterschiedlichen Stromlaufplänen verwendet. Dies mag etwas unstrukturiert aussehen, aber solange die Amis sich nicht nach DIN (Deutschen Industrie Norm) richten, halte ich mich auch nicht daran.

Der Blick ins Datenblatt eines „7400“ [27] ist für die Verwendung des Bausteins unerlässlich. Ganz wichtig sind die zeitlichen Zusammenhänge der logischen Verknüpfungen. Zwischen dem tatsächlichen Umschalten des logischen Eingangszustands und der Änderung des Ausgangspegels verstreicht die sogenannte Verzögerungszeit, auch als Gatterlaufzeit bezeichnet. Im Datenblatt wird zwischen $t_{PLH} = 11 \text{ ns}$ und $t_{PHL} = 7 \text{ ns}$ unterschieden.

Die erste Zeit bezeichnet den Übergang von LOW zu HIGH, die zweite Angabe ist der Wechsel von HIGH zu LOW. Es sind die typischen Zeiten angegeben. Für den Schaltungstechniker ist sehr genau zwischen den „typischen“ und den „maximalen“ Werten zu unterscheiden. Für Aufbauten ohne besondere Anforderungen bezüglich des Temperaturbereichs und bei exakt definierten Versorgungsspannungen ist der typische Wert meist brauchbar.

Sobald aber ein Einsatz über den gesamten Temperaturbereich ansteht, der SN7400 ist von -55 bis $125 \text{ }^\circ\text{C}$ spezifiziert, d.h. MIL für militärische Anwendungen, oder über den zulässigen Spannungsbereich ($5 \text{ V} \pm 10\%$) geplant ist, sind immer die Grenzwerte zu berücksichtigen. Gerne wird beim Testaufbau des Bausteins im Labor oder auf dem Steckbrett der Anschluss der Stromversorgung vergessen. Zudem befinden sich vier Zweifach-NAND in einem Package.

5

Hardware eingebetteter Systeme

Mikrorechner in vielerlei Ausprägungen umgeben das moderne Leben. Bereits wenn der Wecker klingelt, setzt sich in vielen Haushalten das erste elektronische Helferlein in Gang. Beim Griff nach der elektrischen Zahnbürste springt der nächste Mikroprozessor an. Über die restlichen elektronischen Gadgets soll kein weiteres Wort verloren werden – es sind Dutzende.

Mikrocontroller, eine Sonderform des Mikroprozessors, bestimmen unseren Alltag. Deren Funktion zu erkennen, den inneren Aufbau zu verstehen und damit diese Technik beherrschen zu lernen, stellt sich dieses Kapitel zur Aufgabe. Im Laufe der letzten Jahre sind die Leistungen von Mikrocontrollern, kurz MCUs (microcontroller units), in unglaubliche Größenordnungen vorgedrungen, bei gleichzeitig stark fallenden Kosten.

Stichwörter zur Einschätzung dieser Technik sind Busbreite und Taktfrequenz. Unter der Busbreite wird die Datenwortlänge, mit der die CPU (central processing unit) der MCU arbeitet, verstanden. Bis zur Jahrtausendwende waren die sogenannten 8-Bit-CPUs die am häufigsten eingesetzten MCUs. Die seinerzeit sehr stark promoteten 16-Bit-Maschinen konnten sich nicht wirklich durchsetzen. Derzeit (2020) sind 32-Bit-Rechner mit der ARM-Architektur stark verbreitet.

Das Faszinierende bei dieser Architektur ist der historische Kontext. Ursprünglich waren diese Prozessoren als Nachfolger für den Einsatz im Heimcomputer der 1980er-Jahre entwickelt worden. Die damaligen Entwickler wollten einen Leistungssprung und entschieden sich für zwei, damals als durchaus gewagt zu bezeichnende, Schritte. Erstens entwarfen sie, anstelle einer evolutionären Weiterentwicklung der existierenden 8-Bit-CPUs, gleich eine 32-Bit-RISC-Maschine. Der zweite Schritt war dann revolutionär. Sie verzichteten auf eine eigene Fertigung des Siliziumchips und lizenzierten lediglich ihr Design. Dieses Vorgehen wurde seinerzeit als geradezu abenteuerlich angesehen. Kaum jemand glaubte an den Erfolg dieses Unterfangens. Tatsächlich dauerte es bis nach der Jahrtausendwende, bis diese Idee so richtig Früchte trug. Dafür sind die Ergebnisse aber umso beeindruckender. An ARM-Prozessoren und an ARM Cortex M3/4 MCUs ist praktisch kein Vorbeikommen mehr – die Bereitstellung von Intellectual Property war also ein voller Erfolg.



Bild 5.1 Flugmodell ausgerüstet mit einem 32-Bit-ARM7TDMI-Mikrocontroller

Die ARM-CPU basierte auf einem minimalistischen RISC-Befehlssatz (reduced instruction set computer, RISC). Die Anzahl der implementierten Maschinenbefehle wurde im Gegensatz zu einigen anderen Lösungen anderer Marktteilnehmer drastisch reduziert. Damit wurde der Aufbau der CPU vergleichsweise einfach und benötigte deutlich weniger Logik als andere Bausteine. Komplexe Operationen mussten deswegen aus einer Abfolge von Assembleranweisungen programmiert werden. Das freute nicht unbedingt alle Anwender. Die Konkurrenz versuchte mit CISC-Prozessoren (complex instruction set computer, CISC) den missliebigen Spinners aus Großbritannien das Wasser abzugraben. Infolge des einfachen Designs konnte die ARM-RISC-CPU jedoch deutlich schneller getaktet werden. Ein erbitterter Kampf um Marktanteile entbrannte.

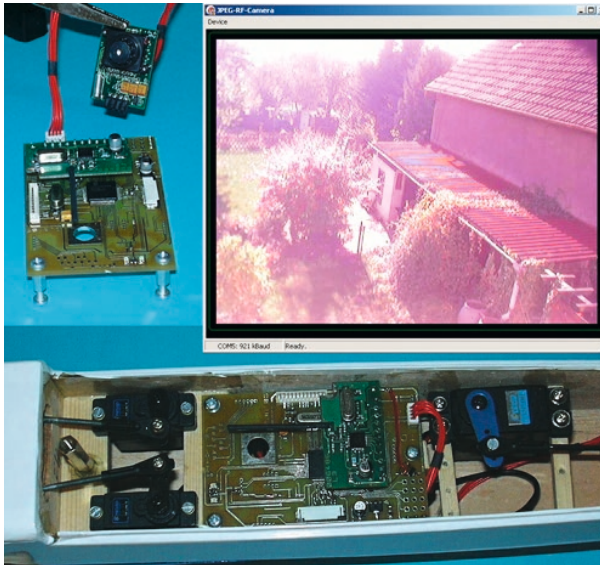


Bild 5.2

Collage aus Bordcomputer, Bildübertragung der Bordkamera und Einbau des Systems im Rumpf des Flugmodells¹

¹ Mein erster Kontakt mit dem ARM7TDMI erfolgte im Zusammenhang mit einem Wettbewerb des US-Magazins „Circuit Cellar“ im Oktober 2005. Bild 5.1 und Bild 5.2 zeigen die Ergebnisse.

Warum gerade die ARM-Architektur derart an Einfluss gewinnen konnte, ist nicht so recht klar. Mit Sicherheit gibt es dafür mehrere Gründe. Die Industrie tat sich am Anfang recht schwer, den Nutzen dieser Rechenzweige zu erkennen. Das Einsatzgebiet der ersten Mikrocontroller waren demzufolge eher einfache Anwendungen, z.B. Tastaturabfragen in Keyboards für PCs, Kassenterminals (point of sale, POS) oder Ähnliches. Als die Automobilindustrie Motorsteuergeräte, Bremsassistenten, elektronische Armaturenbretter etc. – also kurz Bordcomputer im großen Stil – einsetzte, wurden die Helferlein dann richtig populär.

Doch zum Anzeigen von Geschwindigkeit, Drehzahl und Öltemperatur reichen 8-Bit-MCUs völlig aus.² Als jedoch plötzlich Grafikanzeigen erst preiswert und dann weit verbreitet wurden, benötigte man auf einmal Speicher, Speicher, Speicher und Rechenleistung ohne Ende. Die unterschiedlichen Lizenznehmer der ARM-CPU versuchten deswegen, mit der ARM7TDMI-Familie Anfang der Nullerjahre in den lukrativen Markt der eingebetteten Systeme einzusteigen. Der Erfolg war zunächst mäßig. Die Kombination einer leistungsfähigen CPU mit einem höchst mittelmäßigen Interrupt-Controller konnte, insbesondere bei den harten Echtzeitanforderungen eingebetteter Systeme, nur schwer punkten.

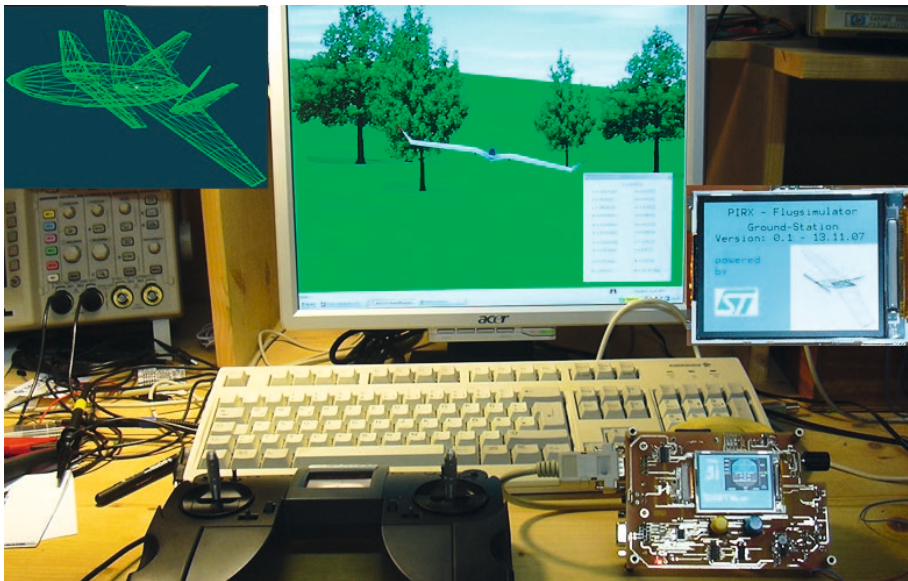


Bild 5.3 Testaufbau eines Steuersystems zur Ankopplung an den Modellflugsimulator Reflex™: Das Steuersystem nutzt einen ARM9 zur Grafikbearbeitung und Animation von Drahtgitter-Modellen³. Es sind zwei Screenshots des TFT-Displays in das Bild montiert.

² Aus sicherer Quelle ist bekannt, dass die Anzeigen des Dashboards eines Porsche Boxster mit einer modifizierten 6502-CPU kontrolliert werden. Für die Nerds unter Ihnen – der C64 lässt grüßen!

³ Mit der Problematik der Flugsteuerungen bzw. -regelungen werden wir uns noch des Öfteren in diesem Buch auseinandersetzen.

Mit der Einführung der Cortex M3/4-Familien wurde dieses Manko behoben. Seitdem erobert die ARM-Architektur eine Bastion nach der anderen im Wettlauf der Systeme. Um hier eine fundierte Einsatzentscheidung treffen zu können, ist die Kenntnis der grundsätzlichen Funktionsweise eines Mikrocontrollers unabdingbar. In den folgenden Abschnitten werden die Grundlagen des Aufbaus von MCUs beschrieben. Infolge der Vielfalt der am Markt erhältlichen Mikrocontrollerfamilien werden manche Funktionen hin und wieder stark vereinfacht vorgestellt. An anderen Stellen wird aber auch konkret auf bestimmte Typen verwiesen.

■ 5.1 Mikrocontroller vs. Mikroprozessor

Der Durchbruch der Digitalrechner ist eng mit dem Aufkommen des ersten vollständig auf einem Siliziumchip integrierten Prozessors verbunden. Die dazu notwendige technologische Entwicklung ist stark mit der Firma Intel verknüpft. In [40] sind dazu ausführliche Informationen zu finden. Der Kniff an der Sache ist die Integration der gesamten Recheneinheit in einem einzigen IC. Die Anforderungen an die Herstellungsprozesse sind enorm. Die Frage, wie viele Logikfunktionen mit welchem Flächenverbrauch auf dem Chip platziert werden können, bestimmt maßgeblich den kommerziellen Erfolg. Eine Zeit lang war Intel diesbezüglich nahezu unangefochten führend (Bild 5.4).

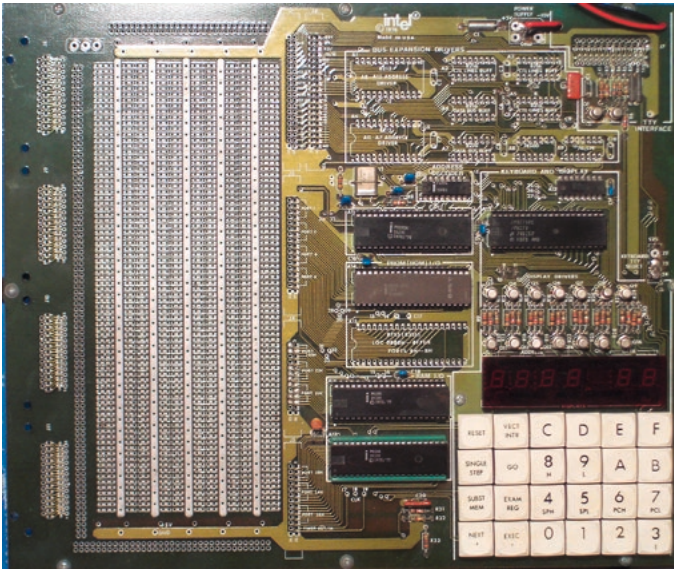


Bild 5.4 Intel-Testsystem MCS-85 aus den 1980er-Jahren mit Prozessor (Mitte), Tastatur, 7-Segment-Anzeige, Peripherie und Erweiterungsfeldern

Der Prozessor allein ist aber nutzlos ohne ein gewisses Maß an Peripherie, als da sind Stromversorgung, Takterzeugung, Speicher und Kommunikationsschnittstellen. Diese bilden in der herkömmlichen Bauweise eines Desktop-PC das Motherboard. Dem Prozessorbauer kann das Design eines speziellen Motherboards streng genommen gleichgültig sein, ihn interessiert nur die CPU selbst.

Damit die Ingenieure und Softwareentwickler arbeiten können, werden neben fertigen Prozessoren auch sogenannte SDKs (software development kit, SDK) angeboten. In vielen Fällen gibt es sie in Form eines „Einplatinencomputers“.⁴ Recht schnell stellte sich heraus, dass man mit den Einplatinenrechnern viele nützliche Dinge tun konnte. Intel brachte dann 1976 mit dem MCS-48 einen 8-Bit-Rechner heraus. Darin waren CPU, Speicher, Timer, Ports und ein Interrupt-Controller in einem IC vereinigt. Obendrein konnte der Speicher unter einer UV-Lampe gelöscht und damit mehrfach beschrieben werden. Der Mikrocontroller war geboren.

■ 5.2 CPU

Die CPU (central processing unit) bildet den Kern eines jedes Mikrocontrollers. Ihre Eigenschaften bestimmen die Leistung des Bausteins. Zum Verständnis des Funktionsprinzips ist zunächst der Befehlszyklus einer CPU von Interesse. Das Steuerwerk einer CPU „versteht“ nur Maschinenbefehle, oft auch als Op-Code oder Assemblerkommando bezeichnet. Die Liste aufeinanderfolgender Op-Codes ist das Programm des Mikrocontrollers. Diese Liste ist im ROM (read-only memory) abgelegt (dazu später mehr).

Jeder (!) Befehlsablauf-Zyklus eines Prozessors erfolgt nach dem Schema in Bild 5.5. Der Programmzähler (PC) adressiert ein besonderes Register im Registerfile der CPU, d. h., er zeigt auf den auszuführenden Befehl. Dieser Befehl bzw. richtigerweise sein entsprechendes Bitmuster wird geladen und decodiert. Gleichzeitig wird der PC inkrementiert. Im Ergebnis der Decodierung des Befehls im Steuerwerk der CPU wird erkannt, ob ein Operand nötig ist. Falls ja, wird dieser eingelesen, ins Operationswerk der CPU transferiert und anschließend der PC wiederum inkrementiert. Damit ist ein Befehlszyklus abgeschlossen, und der nächste Befehl wird ausgeführt. Die Resultate der Berechnungen werden im Flag-Register (xPSR – Prozessorstatus-Register) gespeichert. Die allgemein nutzbaren Register R0 bis R15 befinden sich im Operationswerk der CPU und sind nicht Bestandteil des

⁴ Der ostdeutsche Student träumte nicht von einem MCS-85, sondern von einem LC-80 [94]. Der LC-80 war ein Entwicklungssystem auf der Basis des Z80-Derivats U880. Das Gerät kostete damals um die 990 Mark. Der Student (ohne verlängerten Wehrdienst) erhielt 200 Mark Stipendium (mit zusätzlich 100 Mark von den Eltern war auszukommen). Der BAföG-Höchstsatz 2020 beträgt 861 €. Auf heutige Preise umgerechnet hätte der LC-80 damit ca. 2500 € gekostet. Dafür bekommt man heutzutage beinahe einen gebrauchten Supercomputer.

Index

Symbole

7-Segment-Anzeige 58

A

Abtastrate 141
Abzählanschreibweise 45
Addition 49, 65
aktive Bauelemente 9
Alignment 246
Apollo Guidance Computer 30
Arbeitsgerade 22
Arbeitspunkt 22
arithmetische Grundfunktion 65
atomare Operation 264

B

BCD-Kodierung (binary coded decimal)
50
Bildungsvorschrift 52
Bitzeit 152
Bleisulfid 10
Breakpoints 176

C

Caesar-Chiffriermethode 199
call-by-reference 167
call-by-value 167
Capture/Compare-Einheit 136
Carrington-Ereignis 24
CISC 96

CMSIS 122
Compreter 197
CPU 95, 99
Cross-Compiler 168

D

Datenbus 114
DMA 154
DRAM 110
Drehfeld 212
duale Rechenverfahren 49

E

EAROM 108
EEPROM 108
Einerkomplement 53
Einplatinencomputer 99
Einschaltstrom 7
Elektronenröhre 25
ENIGMA-Chiffriermaschine 199
EPROM 108
Ersatzschaltbild 21

F

Ferritkernspeicher 105
Fertigungsmasken 106
Festkörperschaltung 10
FIFO-Prinzip 125
Flag-Register 99
Flash-ROM 109
FMCW-RADAR 286

Fotowiderstand 235
 Funktionspointer 163
 Funktionspointerliste 180
 fusible link 107

G

Gatterlaufzeit 33
 Gegenkopplung 71
 Geheimtext 200
 General Purpose Timer 136
 Glitch 73
 globale Variablen 165
 Gray-Code 52
 Grenzeffekte 10

H

HAL 122
 Halbaddierer 65
 Halbleiter 9
 Haltepunkt 176
 Hamming-Code 55
 Hamming-Distanz 52
 Hebdrehwähler 61
 HSE 112
 HSE-Quarz 172
 HSI 112
 hybrider Schrittmotor 211
 Hyperraumwürfel 44

I

Impedanz 8
 Induktivität 8
 integrierter Schaltkreis 26
 Interrupt 124
 Interrupt-Event 126
 Interrupt-Handler 127
 Interrupt Service Routine (ISR)
 125
 Invertierung 26

J

JTAG-Adapter 176

K

Karnaugh-Veitch 44
 Kirchhoff'sche Sätze 12
 Klartext 200
 Knotenregel 13
 Kodierung 49
 Komplement 52
 Kondensator 7
 Kontaktprellen 306
 konzentriertes Bauelement 7
 kundenspezifischer ROM 107

L

Ladungspumpe 108 f.
 Latenzzeit 129
 Laufzeit 129
 LDR 235
 linearer Adressraum 104
 Logisim 40
 LTspice 17

M

malloc() 166
 Maschenregel 13
 Maxterm 37
 Methode des „scharfen Hinsehens“
 14
 Mikroschritt 213
 Mikroschritt-Mode 214 ff., 222 ff., 226 ff.,
 231 ff., 236 ff., 247 ff., 253 ff., 258 ff.,
 265 ff., 271 ff., 286, 289, 291, 293 ff.,
 301 ff., 307
 Minterm 37 f.
 Mitkopplung 71
 MMU 166
 Mono-Flop 84
 Multiplikation 49
 Multitasking 179

N

Netzliste 16
 Normalformen 37
 Nulltask 189

O

ODER-Verknüpfung 28
 Ohm'sches Gesetz 6
 Oktalzahl 51
 One-Time-Pad 92
 Open Collector 117
 Operationswerk 99
 Oxymoron 165

P

Parallelbus 114
 Parität 54
 PLL 112
 polyadisches Zahlensystem 51
 POR 104
 power on reset (POR) 78
 Priorität 129
 Programmiergerät 108
 Programmzähler 102
 Prozessorstatuswort 129
 Prüfzeichen 54
 Push-Pull 115

Q

Quantisierung 141
 Quarzoszillator 111

R

RAM 102
 reentrant 132
 Reflektor 201
 RESET-Vektor 103
 RISC 96
 ROM 102f.
 Rotor 200

Rückführung 71
 Rückkopplung 71

S

Schaltfunktion 36
 Schaltnetz 57
 Schaltwerk 71
 Schrittweite 51
 SDK 99
 SFR 122
 Siliziumdioxid 107
 Spannungsquellen 9
 Speicherlayout 102, 165
 SPI 146
 SPICE 17
 SRAM 110
 Stack 165
 Startbit 152
 STARTUP_FROM_RESET 177f.
 static 165
 Stellenschreibweise 46
 Steuerwerk 99
 ST-LINK/V2 251
 Stoppbit 152
 Stromquellen 9
 Stromverstärkungsfaktor 19
 symbolische Methode der Wechselstromtechnik 15
 synchrone serielle Schnittstelle 146

T

Taktflankensteuerung 75
 TCXO 112
 Telegrafenkabel 24
 Telegrafie 24
 Temperaturgang 6
 temporäre Variable 165
 Transistor 18
 Tri-State-Stufe 119
 TTL-NAND-Gatter 32
 TTL-Technologie 30

U

UART 152
UEXEC 183
Umkehrwalze 201
UND-Verknüpfung 27

V

Vektor 103
verbotene Zustände 90
Verzögerungszeit 33
Volladdierer 65
Volldisjunktion 37
Vollkonjunktion 37
Vorteiler 136

W

Wahrheitstabelle 36
Wahrheitstafel 36
Wellenwiderstand 115
Wichtung 51
Widerstand 5
Widerstandskoeffizient 6

Z

Zeichengenerator 106
Zugriffzeit 106
Zweierkomplement 54
Zweipol 9, 12