

Digitaltechnik

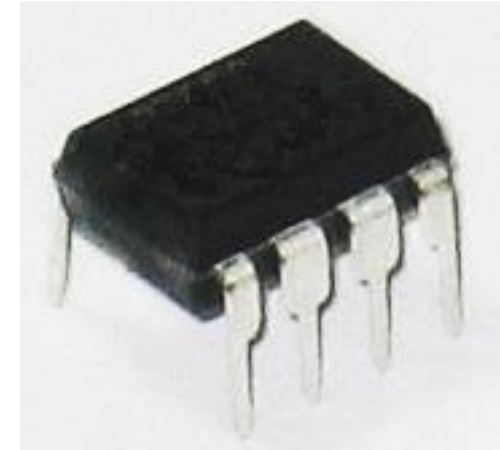
Kapitel 7, Realisierung digitaler Systeme

Prof. Dr.-Ing. M. Winzker

*Nutzung nur für Studierende der Hochschule Bonn-Rhein-Sieg gestattet.
(Stand: 21.03.2022)*

7.1 Diskrete Logikfunktionen

- Es existiert eine große Auswahl an Bausteinen mit einfachen Funktionen:
 - Logikfunktionen: AND, OR, NAND, NOR, ...
 - Flip-Flops, Taktteiler,
 - Zähler, Decoder für 7-Segment-Anzeigen, ...
- Die Bausteine haben eine 4-, oder 5-stellige Bezeichnung, beginnend mit „74“



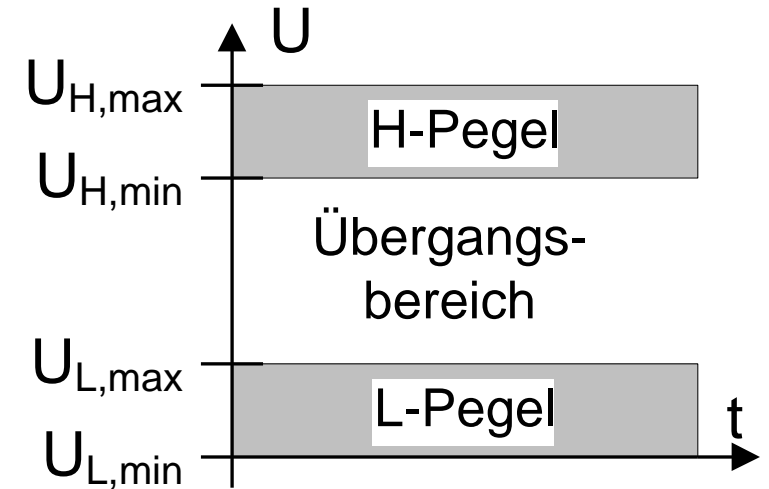
Es gibt verschiedene Baustein-Familien, u.a.

- **Standard-TTL** (Transistor-Transistor-Logik): Familien-Bezeichnung: 7400
- **Schottky-TTL** (74S00): Sehr geringe Schaltzeiten, sehr hohe Leistungsaufnahme
- **Low-Power-Schottky-TTL** (74LS00): Gute Schaltzeiten, geringe Leistungsaufnahme
- **High-Speed-CMOS** (74HC00): CMOS-Technologie, andere Eingangspegel
- **High-Speed-CMOS, TTL-Pegel** (74HCT00): CMOS mit TTL Pegeln
- Zusätzlicher Prefix gibt Hersteller an: SN: Texas Instruments, DM: Fairchild, ...

Pegelbereiche

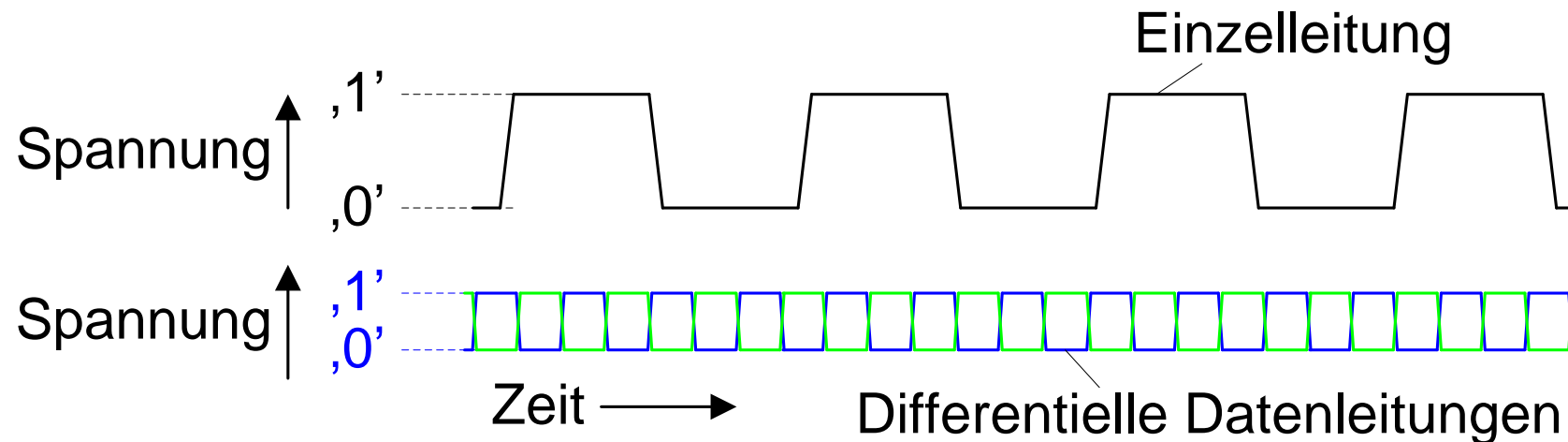
- Für 0 und 1 werden **Spannungswertebereiche** mit den Bezeichnungen L („Low“) und H („High“) verwendet
(Genaue Spannungswerte wären ohnehin nicht exakt einzuhalten)
- Zwischen den Wertebereichen ist ein **Übergangsbereich**, in dem das Signal **undefiniert** ist
- Auswahl an typischen Werten für Eingangspegel:

Standard	TTL	CMOS	LVTTL	ECL
$U_{H,max}$	5 V	5 V	3,3 V	0 V
$U_{H,min}$	2,0 V	3,5 V	2,0 V	-1 V
$U_{L,max}$	0,8 V	1,5 V	0,8 V	-1,4 V
$U_{L,min}$	0 V	0 V	0 V	-5V



Differentielle Datenübertragung

- Bei der differentiellen Datenübertragung werden zwei Leitungen verwendet, die entgegengesetzte Spannungspegel einnehmen
 - „LVDS“ (engl. „Low Voltage Differential Signaling“)
 - Vorteil: Die Datenübertragung ist weniger störanfällig und kann mit kleineren Spannungen und wesentlich schneller erfolgen
- Verwendung in USB, S-ATA, HDMI, ...



7.2 Komponenten

In Kapitel 1 wurden die verschiedenen Möglichkeiten zur Implementierung einer Digitalschaltung vorgestellt:

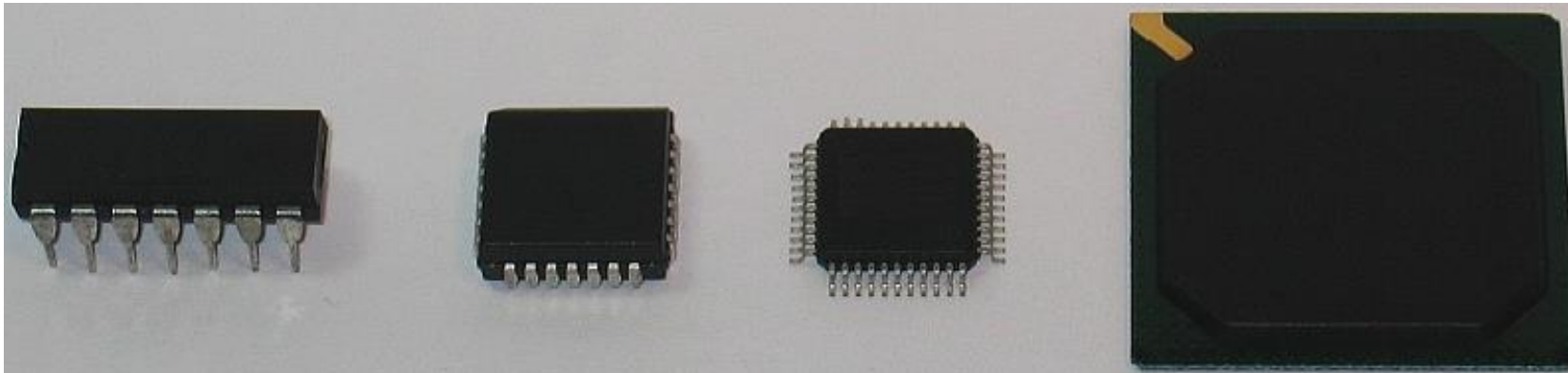
- Standardbauelemente, ASSP
- Kundenspezifische Schaltung, ASIC
- Programmierbare Schaltung, FPGA
- Mikrocontroller, μC

Vorteile integrierter Schaltungen

- Geringe Baugröße: Nur ein Gehäuse erforderlich
- Geringere Kosten: Ein Gehäuse, kleinere Platine, einfachere Fertigung
- Höhere Geschwindigkeit: Sehr kurze Verbindungsleitung zwischen Transistoren
- Geringe Parameterabweichungen: Alle Transistoren einer Schaltung wurden in gleicher Weise gefertigt

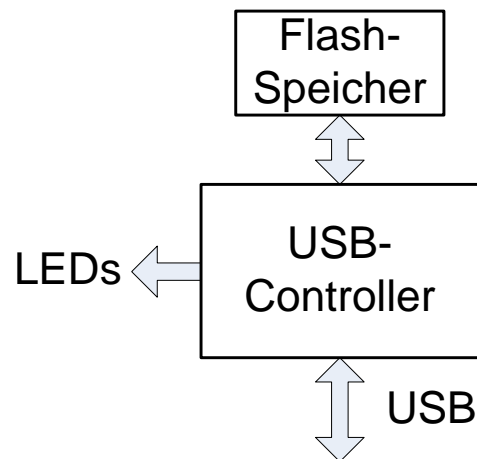
Standardbauelemente – ASSP

- Integrierte Schaltungen für häufig benötigte Funktionen
 - Mikrocontroller
 - CPU (Intel Core-i, AMD Opteron, ...)
 - Grafikcontroller (Radeon, Geforce, ...)
 - DRAM Speicher
- Komplette Schaltung auf einem einzigen Stück Halbleiter
 - Über 1 Milliarde Transistoren auf etwa 0,25 bis 2 cm² Fläche möglich
 - Bezeichnung: **ASSP** – „**A**pplication **S**pecific **S**tandard **P**roduct“

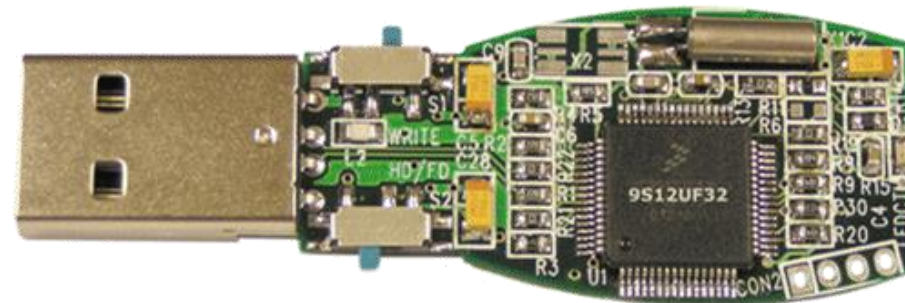


ASSP: Controller für USB-Stick

- Kompletter USB-Stick besteht hauptsächlich aus zwei ICs:
 - NXP Controller MC9S12UF32
 - NAND-Flash
 - Separater Baustein, da spezielle Technologie für Flash-Speicher
 - Zusätzlich Widerstände, Kondensatoren, LED, Quarz, ...
- Referenzdesign auf:
<http://www.nxp.com/pages/usb-thumb-drive-reference-design:RDHCS12UF32TD>
- Kosten für aktuelle Controller: ca. 1 Euro (geschätzt aus Preis für USB-Stick)



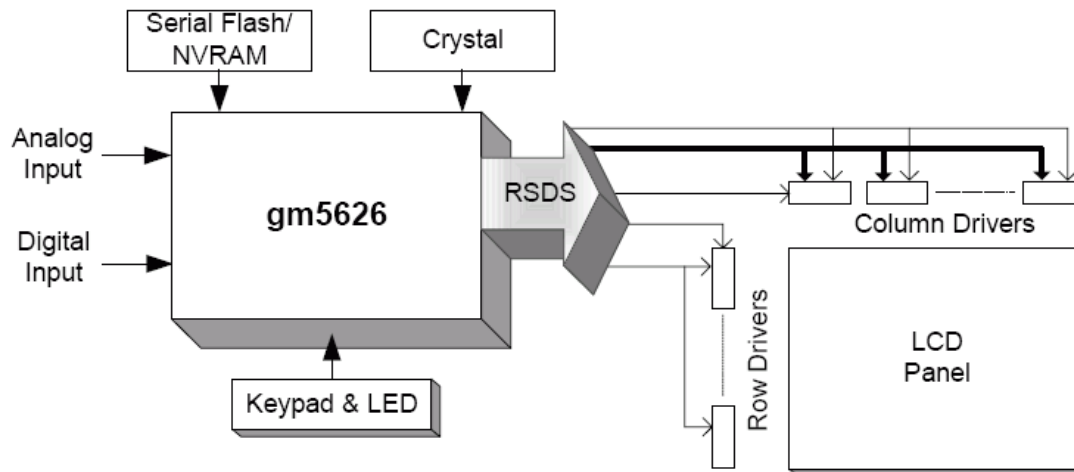
(Quelle: Freescale,
jetzt NXP)



Flash-Speicher auf Rückseite der Platine

ASSP: Controller für LCD-Monitor

- STMicroelectronics gm5626
 - Komplette Signalverarbeitung und Steuerung auf einem Chip
 - Analog- und DVI-Eingang
 - Erkennung des Eingangssignal, Skalierung, Verbesserung der Schärfe, OSD (On-Screen-Display)
 - Steuerung durch eingebetteten Mikrocontroller (Signalverarbeitung und OSD)
 - Direkte Ansteuerung der Treiber für LCD-Panel
- Externes Flash für Logo und OSD des Monitor-Herstellers und Mikrocontroller-Code
- Kosten: ca. 5 Euro (geschätzt, abhängig von Stückzahl)



(Quelle: Genesis Microchip,
jetzt STMicroelectronics)

Kundenspezifische Schaltung – ASIC

- Entwurf eines eigenen ASICs
 - Die hohe Komplexität macht den Entwurf sehr komplex
- Bei 1 Milliarde Transistoren kann nicht jeder Transistor einzeln entworfen werden
 - Entwurf durch spezielle **Hardwarebeschreibungssprache** VHDL (oder Verilog)
 - Große Teams, z.B. 30 Ingenieure und Informatiker, teilweise an mehreren Standorten
 - Gesamte Projektzeit etliche Monate, teilweise Jahre

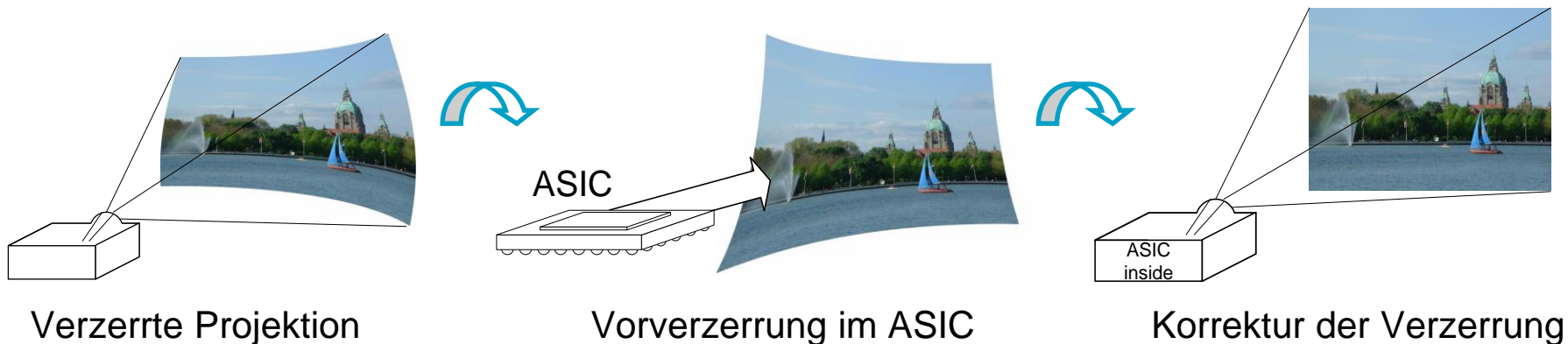
Entwicklungsrisiko

- Hohe **Investitionen** für Personal, Entwicklungssoftware, Prototypenfertigung (NRE, „Non Recurring Engineering“)
 - Allein Prototypenfertigung ab 100.000 € bis über 1 Mio. €
- **Entwurfsrisiko**: Fehlerbeseitigung erfordert oft neue Prototypenfertigung, also neue NRE-Kosten und Monate Verzögerung
- Produktion erfolgt in Chargen zu mehreren tausend bis zehntausenden ICs

ASIC: Controller für „Beamer“

Controller für Daten- und Video-Projektor

- Am Markt verfügbar: Controller für LCD-Controller, ähnlich gm5626 (siehe oben)
 - Aber: Keine Korrektur für „Keystone“ enthalten (Verzerrung bei schräger Projektion)
 - Ziel: Marktvorteil durch besonderen Controller
 - Korrektur von Keystone und Projektion auf gekrümmte Flächen
 - Eigenschaften vorhandener Controller (Signalverarbeitung, OSD, CPU)
- ➔ Eigenentwicklung eines Controller-ASIC
- Entwurf in VHDL, Fertigung in 0,18µm CMOS bei LSI Logic
 - Projektdauer ca. 2 Jahre, 5 ASIC-Entwickler



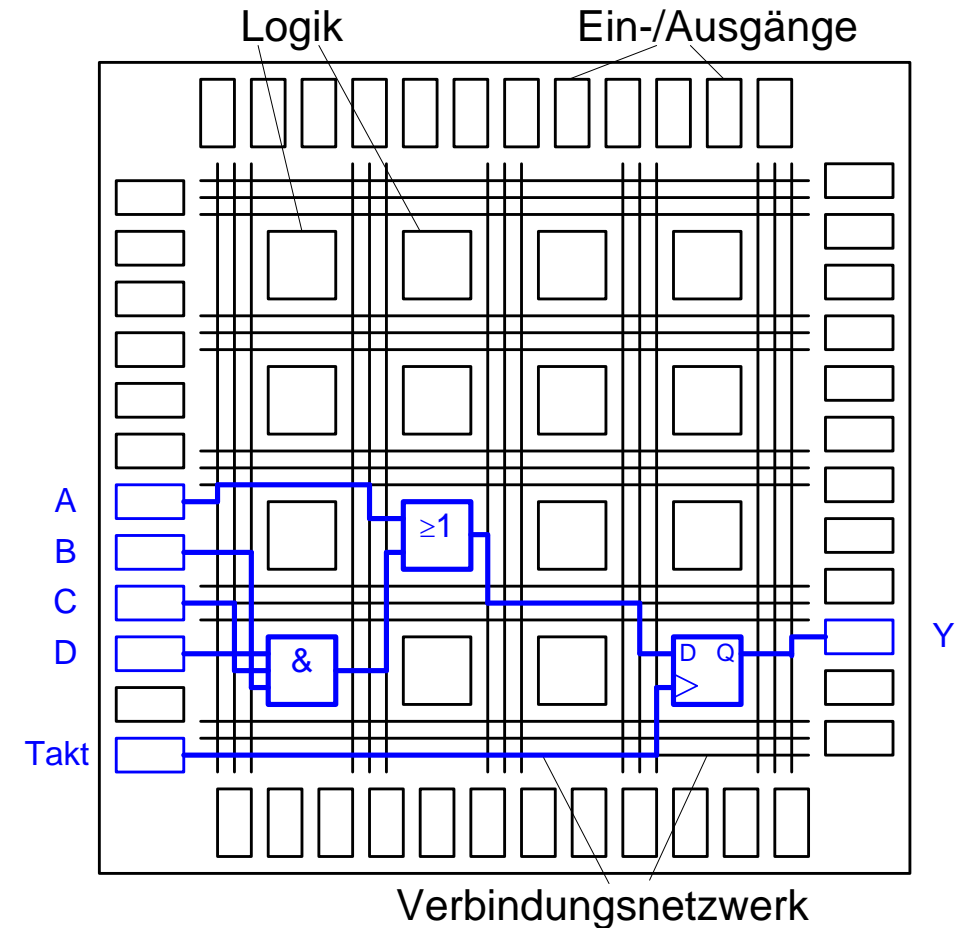
Programmierbare Schaltung – FPGA

Ein „Field-Programmable-Gate-Array“ ist ein Mikrochip mit einer programmierbaren Digitalschaltung

- **Field-Programmable:** Im Einsatzfeld (auf der Platine) programmierbar
- **Gate-Array:** Feld von Gattern mit
 - mehrere tausend Logikfunktionen (wählbar als UND, ODER, EXOR, ...)
 - mehrere tausend Speicherelemente (Flip-Flops, FFs)
 - Verbindungsnetzwerk
- Funktion und Verdrahtung ist programmierbar (**blau**)

Vor- und Nachteile:

- Leistungsfähigkeit fast wie ASIC
- Keine aufwändige Fertigung nötig
- Aber Stückpreis höher als ASIC



Vereinfachte Darstellung: Reale FPGAs enthalten wesentlich mehr Logikblöcke und Ein-/Ausgänge

FPGAs und Mikrocontroller

- FPGAs haben eine andere Funktionsweise als Mikrocontroller
 - **FPGA:** Eine Funktion wird **direkt als Schaltung** umgesetzt
 - **Mikrocontroller:** Eine Funktion wird **Schritt für Schritt** in der CPU berechnet
- FPGA und Microcontroller können prinzipiell für ähnliche Aufgaben eingesetzt werden, haben aber andere Eigenschaften
 - Vorteile FPGA:
 - Deutlich höhere Geschwindigkeit, denn eine Funktion muss nicht Schritt für Schritt berechnet werden
 - Parallelverarbeitung durch viele Funktionsblöcke möglich
 - Hohe Datenrate möglich
 - Z.B. für Videosignale (HDTV: 1280×720 Pixel x 50 Bilder/Sekunde = 46 Mpixel/Sek)
 - Vorteile Microcontroller:
 - Einfacher zu programmieren
 - Sehr flexibel
 - Z.B. für Steuerungen oder Audiosignale (44 kSample/Sekunde)

Vergleich der Alternativen

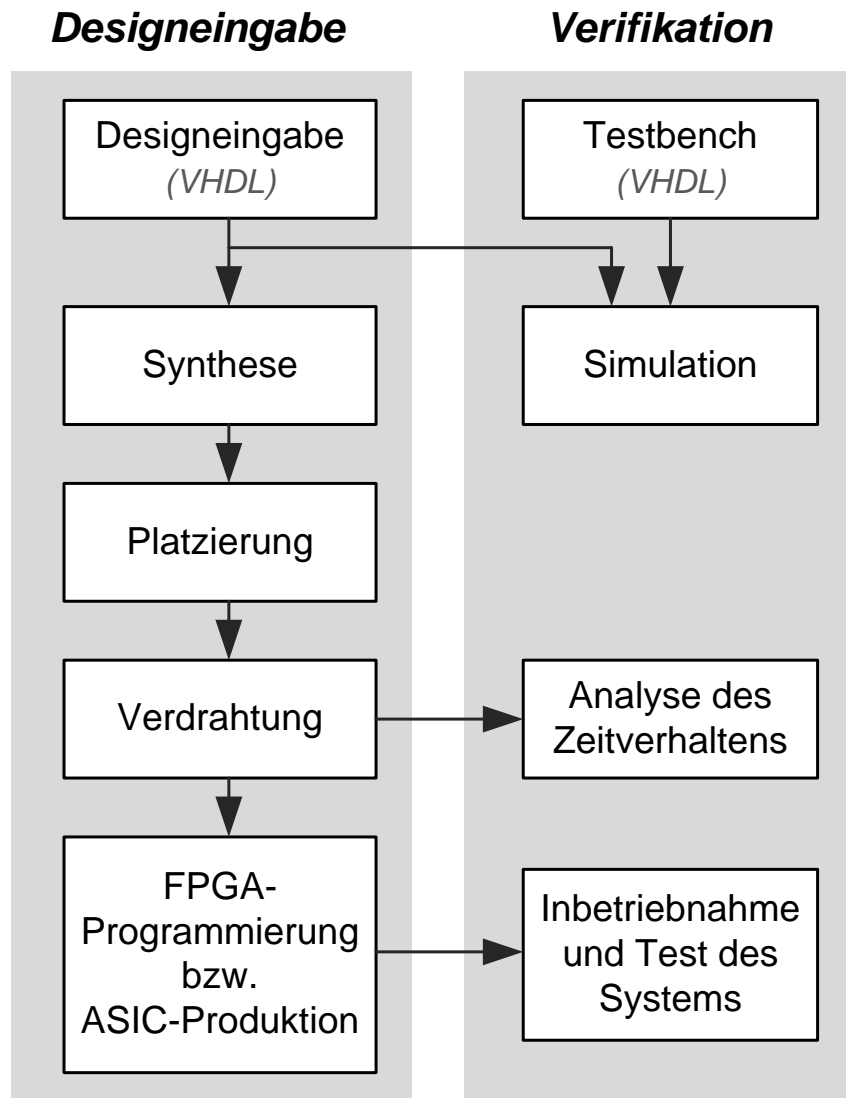
	ASIC	ASSP	µC	FPGA
Hohe Flexibilität	+	—	+	+ +
Geringe Entwicklungszeit	— —	+	+ +	0
Geringe Entwicklungskosten	— —	+	+ +	0
Geringe Stückkosten	+ +	+	+ +	0
Rechenleistung	+ +	+ +	0	+
Verlustleistung	+ +	+ +	0	0
Geringe Stückzahlen möglich	— —	+ +	+ +	+ +
Hohe Stückzahlen möglich	+ +	+ +	+ +	+

- Wahl einer Alternative ist von Randbedingungen eines Projektes abhängig
 ➔ Ausführliche Analyse nötig

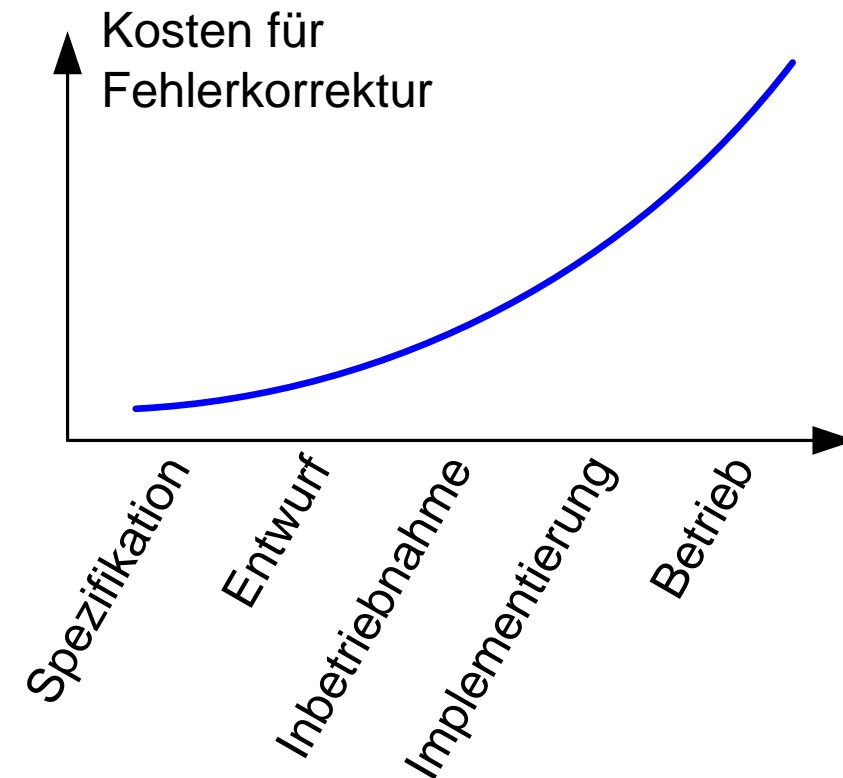
7.3 Verifikation

- Mit **Verifikation** bezeichnet man in der Digitaltechnik die Überprüfung, ob ein Schaltungsentwurf die spezifizierten Eigenschaften erfüllt
 - **Verifikation** bezieht sich also auf den korrekten **Entwurf**
 - Bezeichnung in der Informatik auch „Validation“
 - Mit **Test** wird dagegen die korrekte **Fertigung** einer Schaltung bezeichnet
 - ➔ **Verifikation** erfolgt (prinzipiell) einmalig in der Entwicklung
 - ➔ **Test** erfolgt mit jeder Schaltung nach der Fertigung
- Verifikation ist ein entscheidender Schritt vor Freigabe eines Produktes
 - Bis zu 70% der Entwurfsarbeit komplexer Digitalschaltungen geht in die Verifikation
 - Unzureichende Verifikation kann teuer und sehr imageschädigend sein
 - 1994: Intel Pentium mit Bug beim FDIV-Befehl
 - 2004: Siemens warnt vor Hörschäden durch Handy-Ausschaltmelodie
 - 2016: Hypervisor-Busting Bug in AMDs Piledriver CPUs
 - 2016: Skylake Bug

Verifikation im Design-Flow

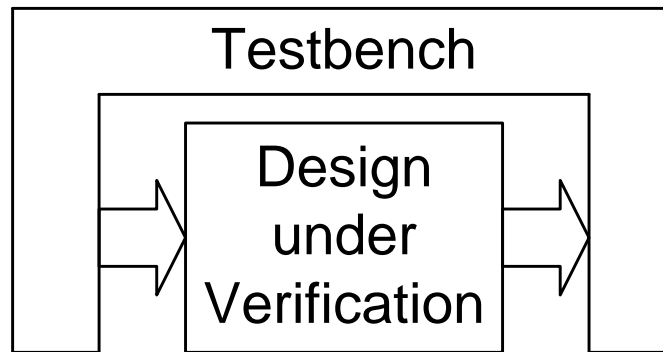


- Die Verifikation begleitet den gesamten Entwurfsablauf
- Je später ein Fehler erkannt wird, um so höher sind die Kosten für eine Korrektur



Schaltungssimulation

- Schaltungssimulation ist eine wichtige Verifikationsmethode
- Eine Schaltungssimulation erfolgt üblicherweise durch eine **Testbench** („Bench“ = Werkbank)
- Die zu verifizierende Schaltung wird als „Design under Verification“ bezeichnet, manchmal auch, nicht ganz korrekt „Device under Test“
- Die Testbench legt Eingangssignale („**Stimuli**“) an die Schaltung an und empfängt die Ausgangssignale („**Response**“)



- Die Testbench kann, wie die Schaltung, in VHDL geschrieben werden
- Testbench und Schaltung werden im Rechner simuliert
 - Rechenplattform ist PC oder Workstation mit Windows oder Unix/Linux

Überprüfung der Simulationsergebnisse

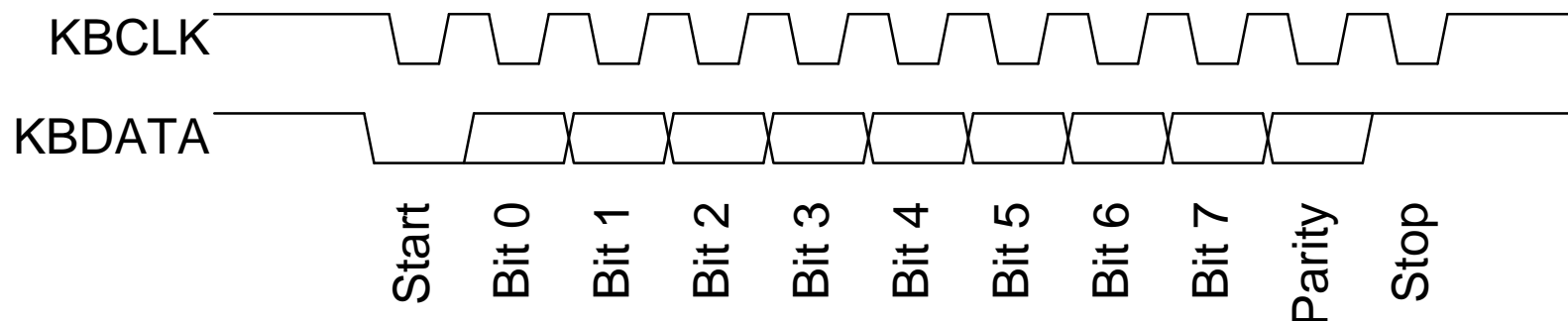
- Für einfache Schaltungen können die Simulationsergebnisse vom Entwickler im „Waveform-Viewer“ überprüft werden
- Dies ist jedoch fehleranfällig, da durch Routine die menschliche Aufmerksamkeit nachlässt
- Eine „**Self Checking Testbench**“ erzeugt die Stimuli und überprüft die Response
- Dazu muss die Testbench die erwartete Response kennen. Dazu gibt es verschiedenen Möglichkeiten:
 - Die erwartete Response wird vom Entwickler in die Testbench programmiert
 - Für einfache Response geeignet
 - Ein zweites Modell („Referenzmodell“) erzeugt die erwartete Response
 - Das zweite Modell kann in VHDL oder einer anderen Sprache (z.B. „C“) geschrieben sein
 - Beide Beschreibungen überprüfen sich gegenseitig
 - Komplexe Algorithmen werden ohnehin vor der VHDL-Implementierung simuliert

Beispiel: PS/2-Tastatur

Design under Verification: Das PS/2-Signal einer PC-Tastatur soll verarbeitet werden

Testbench: Erzeuge das Signal einer PS/2-Tastatur

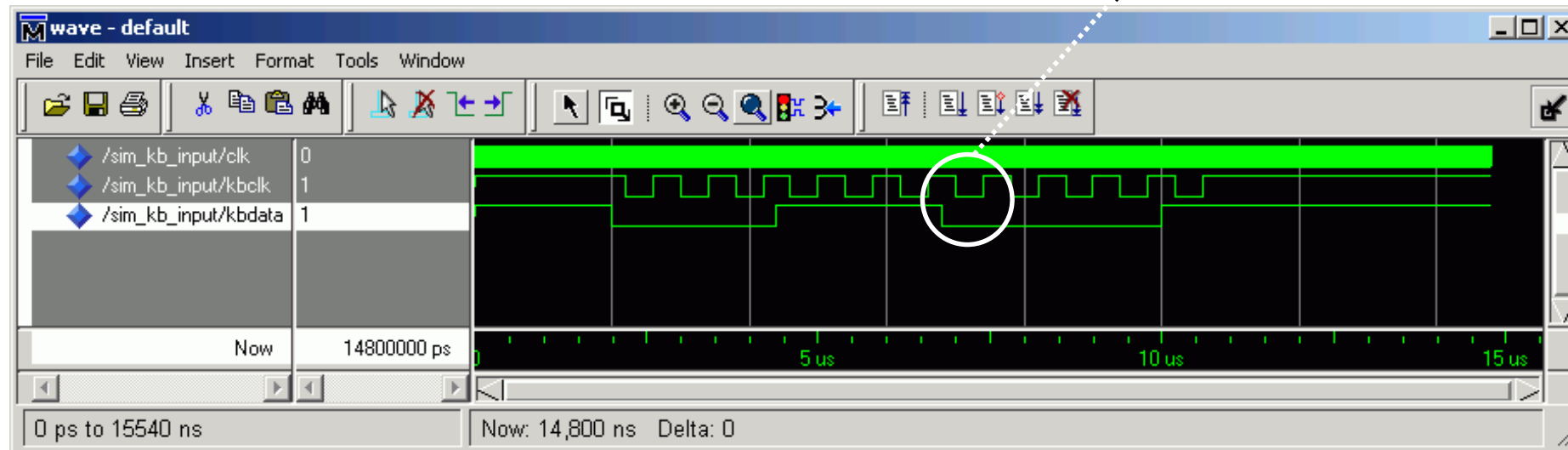
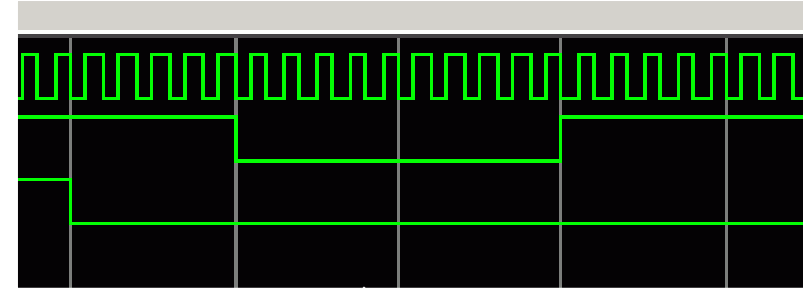
- PS/2: Serielle Datenübertragung über zwei Leitungen
 - **Datenleitung** (KBDATA): Überträgt Informationen
 - **Taktleitung** (KBCLK): Signalisiert gültige Bits
 - Tastendrücke werden durch ein (oder mehrere) Byte dargestellt
 - Taste „A“ = 0x1C; Taste „→“ = 0xE0, 0x74
 - Stecker mit 6 Stiften: Daten, Takt, 5V, Masse, 2 Stifte unbenutzt
- Testbench erzeugt zusätzlich Verarbeitungstakt sowie Reset-Signal



Beispiel: PS/2-Tastatur (II)

Übertragung bei Tastendruck ‚A‘:

- Startbit ‚0‘
- Datenwort: 0x1C = „0001 1100“ (LSB first)
- Parity: ‚0‘ (odd parity)
- Stopbit ‚1‘



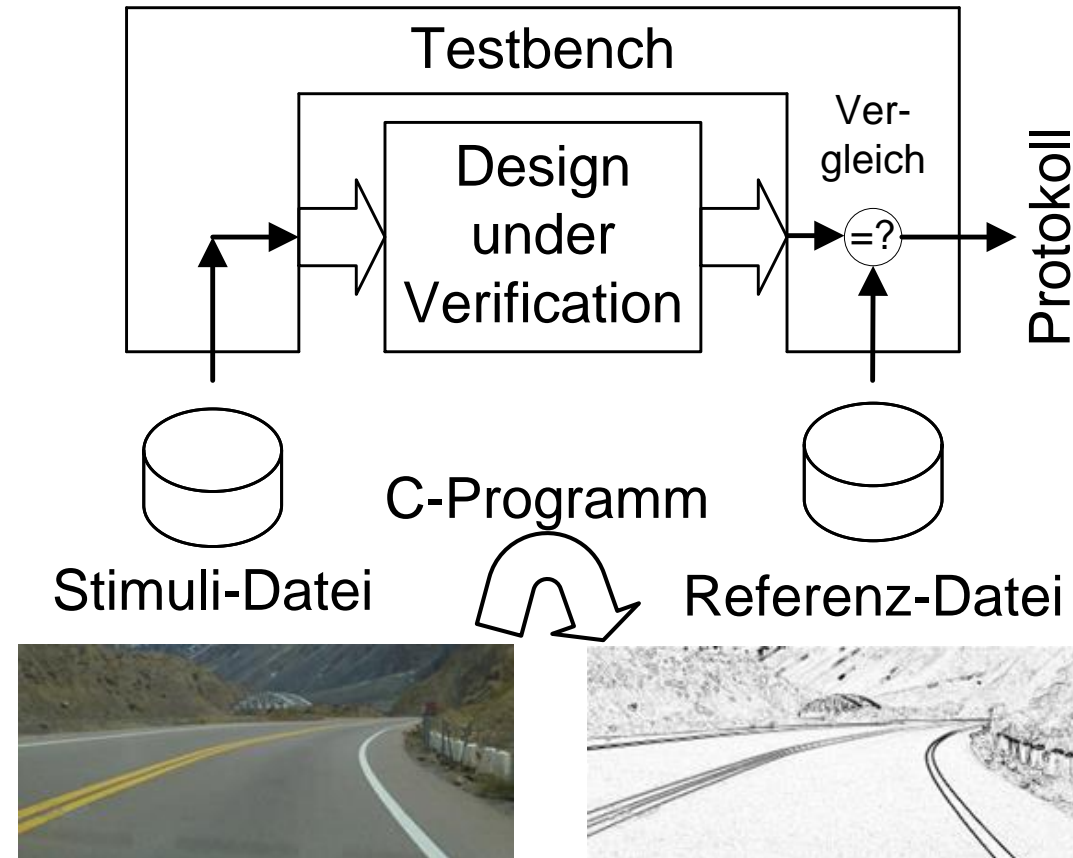
Ausführliches Video unter: <https://youtu.be/6hZc5XQTpzA>

Praktikum: Self Checking Testbench

- Zur Verifikation liest die Testbench Stimuli und Referenz-Response aus Dateien

Beispiel: Kantenfilter für Videodaten

- Der Algorithmus wird mit einem C-Programm entwickelt und mit einigen Testbildern überprüft
 - Die begrenzte Wortlänge in einer Digitalschaltung wird berücksichtigt
- Die Testbench liest ein Testbild und legt es als Stimuli an das Design
- Die Testbench vergleicht die Response des Designs mit dem Referenzergebnis des C-Programms



7.4 Trends in der Mikroelektronik

Für Technologietrends werden meist folgende Bausteine betrachtet:

- **CPU für PC:** Über viele Dekaden leistungsfähigste Bauelemente, Rechenleistung durch Benchmarks (relativ) gut vergleichbar
- **Grafikcontroller, FPGAs:** Hohe Transistorzahlen; durch reguläre Struktur gut skalierbar (also in der Größe veränderbar)
- **DRAM:** Spezielle Speichertechnologie
- **Flash-Speicher:** Spezielle Speichertechnologie

Die Trends für diese „Leuchttürme“ zeigen die Möglichkeiten für alle Anwendungen

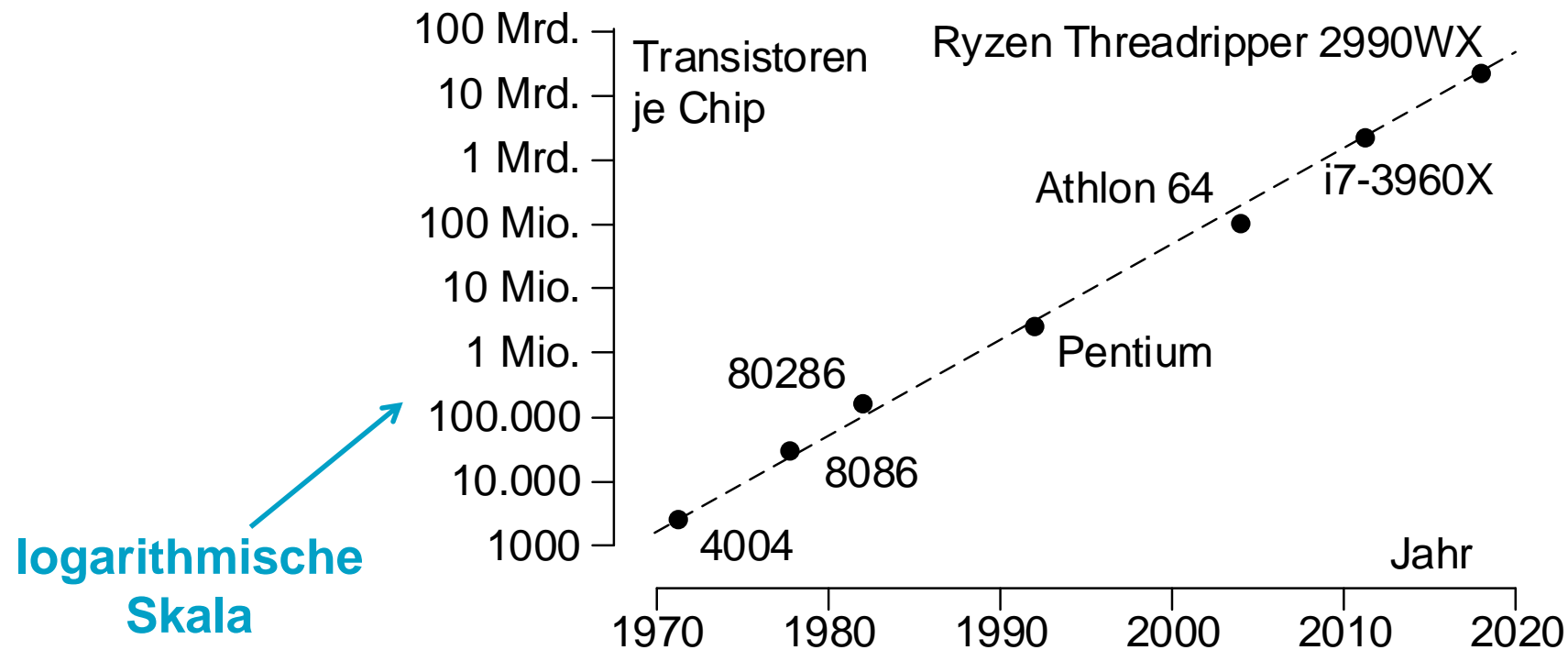
- ➔ Nicht jede Schaltung muss die Möglichkeiten ausnutzen
- ➔ Trends der Vergangenheit setzen sich oft für die nähere Zukunft fort
 - Ausnahmen bestätigen die Regel

Bekanntester Technologietrend: Moore'sches Gesetz

Das Moore'sche Gesetz („Moore's Law“) besagt:

Die Anzahl der Transistoren pro integrierter Schaltung verdoppelt sich alle zwei Jahre

- Trend: Steigende Integrationsdichte
- Bausteine: CPU für PC



Moore'sches Gesetz (III)

- Ein exponentieller Anstieg ist schwer vorstellbar
 - Klassisches Beispiel: Reiskörner auf Schachbrett

Zahlenbeispiel

- Intel 8086: „16-Bit-Mikroprozessor [...] Urvater der 80x86-Familie“
 - 29'000 Transistoren im Jahr 1978

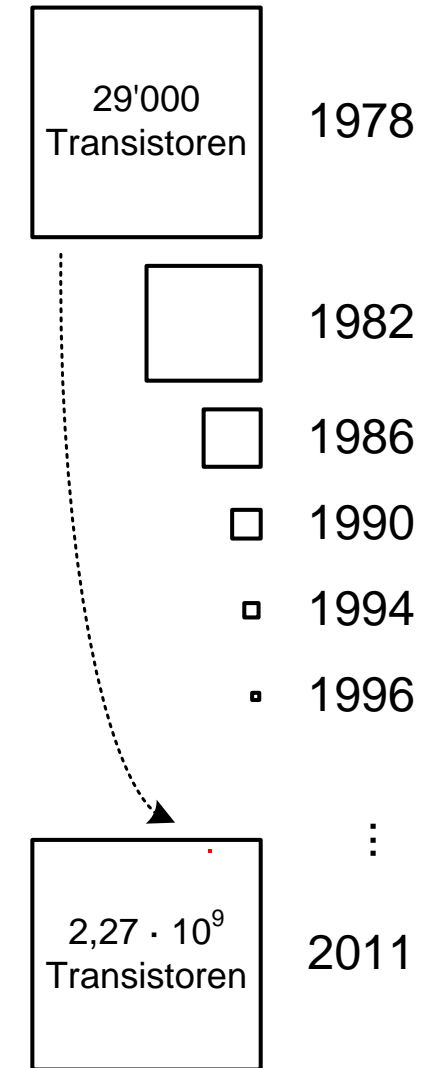
Quelle: http://de.wikipedia.org/wiki/Intel_8086

- Core i7 Sandy Bridge E: Sechskernprozessor
 - 2,27 Milliarden Transistoren im Jahr 2011

Quelle: <http://de.wikipedia.org/wiki/Intel-Core-i-Serie>

- Faktor 78'000 in 33 Jahren
- Chip aus 1978 belegt auf 2011er IC: 0,0013%

Frage: Wie groß ist die Steigerungsrate je 2 Jahre?



Trend: Entwurfsproduktivität

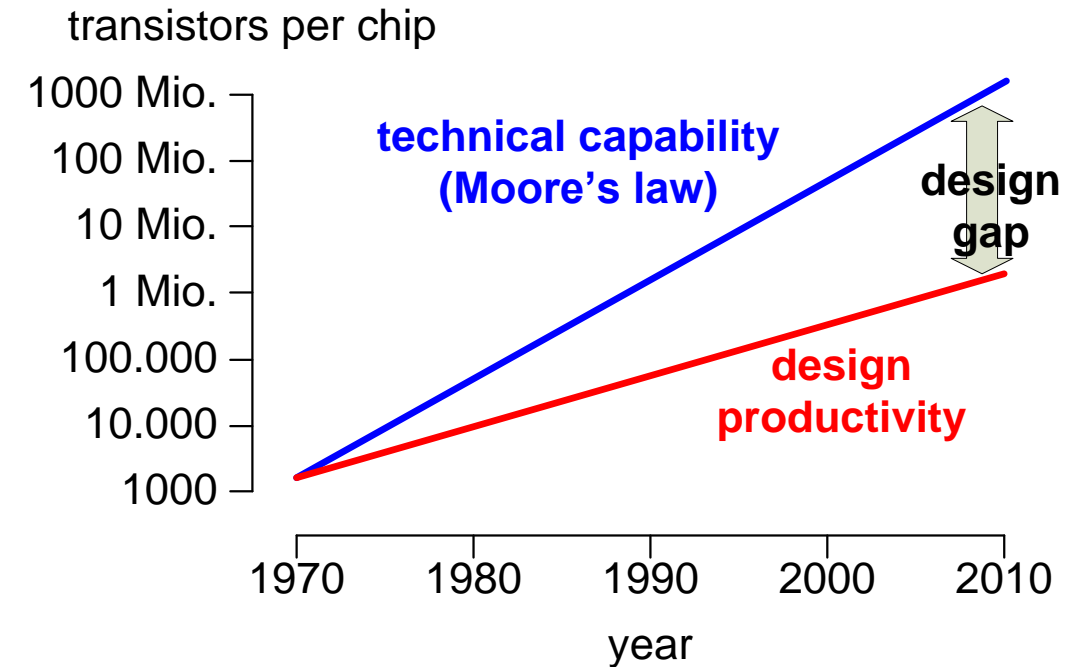
Die Produktivität im Chip-Design steigt langsamer als die technologischen Möglichkeiten

- “Design Gap”

Möglichkeiten zur Erhöhung der Entwurfsproduktivität

- Neue Entwurfsmethodik
 - Schematic entry → VHDL → High-Level Design (Matlab, C)
- Verwendung vorhandener Module
 - Re-use eigener Module
 - Multi-core CPU
 - Lizenzierung fremder Module (IP, Intellectual Property)
 - ARM CPU in Smartphones

➔ Ähnliche Entwicklung für Software: Assembler → Hochsprache → Objektorientierung



Konsequenzen der Technologietrends

Moore'sches Gesetz: Je Chip kann mehr Logik implementiert werden

- 8-bit CPU → 16-bit CPU → 32-bit CPU → 64-bit CPU → Multi-Core CPU
- 80386 CPU hatte separaten Floating-Point Coprozessor 80387
- Athlon 64 und Core i7 haben "Northbridge" integriert

Kosten: NRE Kosten für ASICs steigen

- Mindeststückzahl für ASICs steigt von „tausende“ auf „hunderttausende“

Entwurfsproduktivität: Potential moderner ASIC Technologie schwer nutzbar

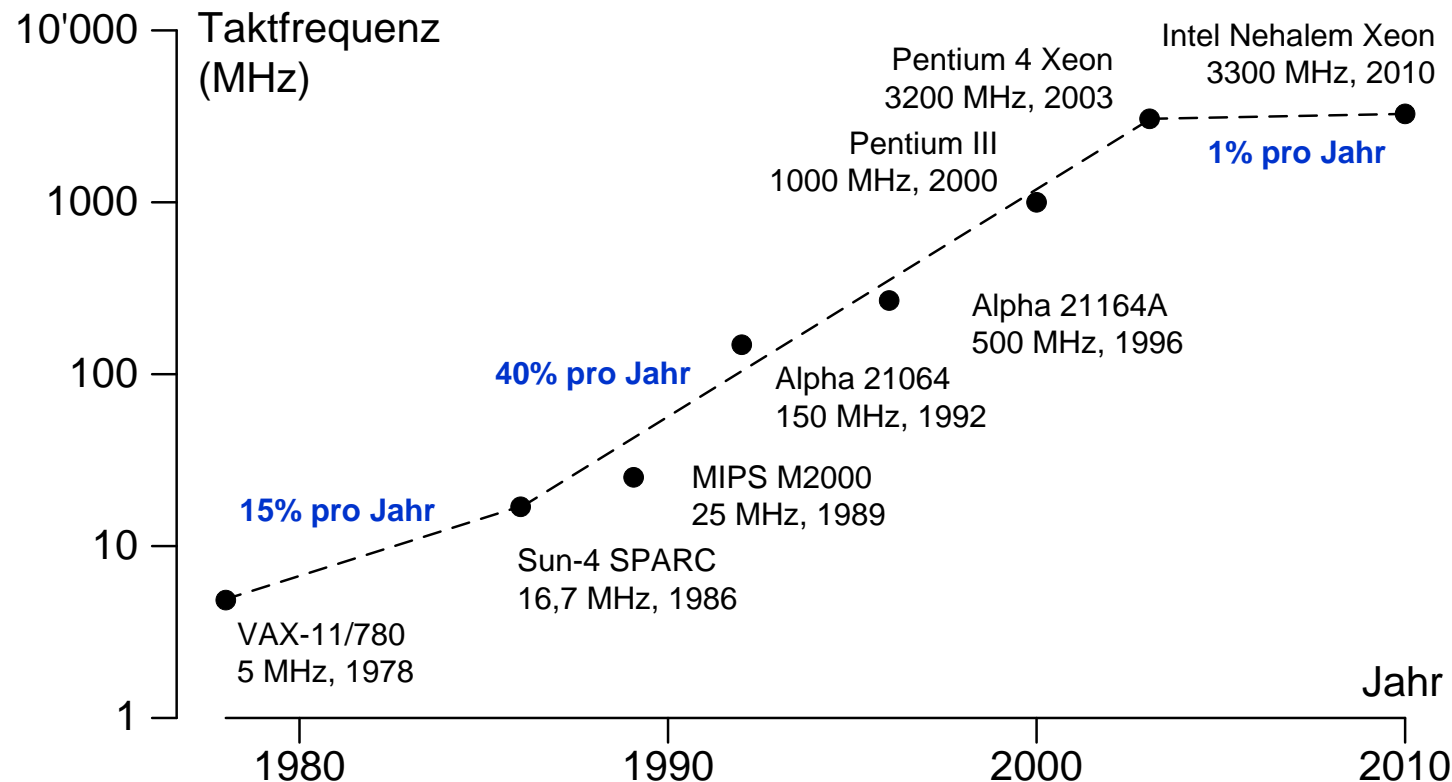
- Welche Anwendung benötigt 1 Milliarde Transistoren?
 - Cache einer CPU lässt sich einfach skalieren
 - Multi-Core CPU kopiert vorhandene CPU-Struktur
- Für welche Anwendung können hunderttausende ASICs verkauft werden?

Konsequenzen

- Einsatz von immer mehr programmierbaren Bauelementen (FPGAs, Mikrocontroller)
 - ➔ Abtausch von Transistoren gegen NRE-Kosten und Entwicklungszeit

Trend: Taktgeschwindigkeit

- Starker Anstieg in den 1990er
- Technisch/ökonomisches Limit bei ca. 4 GHz
 - ➔ Höhere Frequenzen sind möglich (Übertaktung, Kühlung), aber zu aufwändig

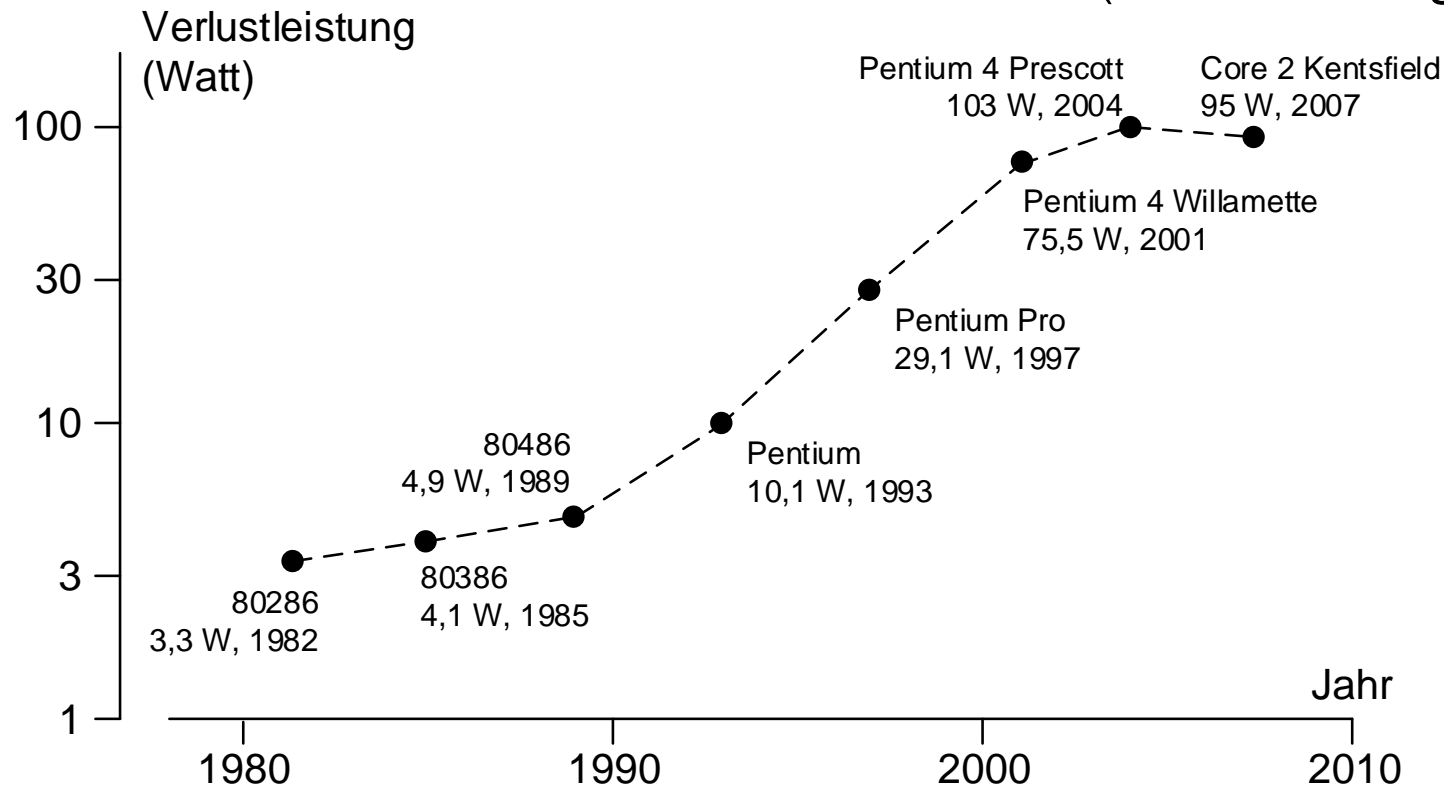


Datenquelle: J. Hennessy, D. Patterson,
Computer Architecture, Morgan Kaufmann, 2012.

Trend: Verlustleistung

- Intel x86 Architektur
- Starker Anstieg um das Jahr 2000
 - ➔ „Power Wall“ bei ca. 100 Watt

➔ Auch das Moore'sche Gesetz wird irgendwann an seine Grenze stoßen
(laut Vorhersagen ca. 2028)



Datenquelle: D. Patterson, J. Hennessy,
Computer Organization and Design, Morgan Kaufmann, 2009.

Verlustleistung als Entwurfsziel

Aus mehreren Gründen ist die Verlustleistung heute ein Entwurfsziel

- Anschaffungskosten für Gehäuse und Kühlung
- Betriebskosten für Energie und Kühlung
- Laufzeit mobiler Geräte, also Laptop, Tablet, Smartphone, ...
- Autarke Sensoren mit langlebiger Batterie oder Energy Harvesting („Smart Dust“)

Video: FPGA Vision - Low-Power Design <https://youtu.be/CU7x892KGJs>

Hintergrundinformationen:

- Google container data center tour, Google Efficient Data Centers Summit, Mountain View, CA April 1, 2009, <http://www.youtube.com/watch?v=zRwPSFpLX8I>
- R. Courtland, "The high stakes of low power," IEEE Spectrum, 2012.
- D. Schneider, "Under the Hood at Google and Facebook," IEEE Spectrum, 2011.

7.5 Wärmeleitung und Kühlung

Zum Nachlesen: S. Maniktala, „Switching power supplies A to Z,“ Newnes, 2012, Kapitel 11, „Thermal Management“

- Die Verlustleistung einer Schaltung wird über Wärmeleitung an die Umgebung abgegeben
- Die Temperatur der Schaltung ergibt sich aus
 - Verlustleistung P („Power“)
 - Thermischer Widerstand R_{th} („Thermal Resistance“)
 - Umgebungstemperatur T_A („Ambient Temperature“)
- Temperaturdifferenz $\Delta T = P \cdot R_{th}$
- Temperatur $T = T_A + \Delta T = T_A + P \cdot R_{th}$
- Das Verhalten ist analog zum Ohmschen Gesetz mit
 - Strom \Rightarrow Verlustleistung
 - Widerstand \Rightarrow Thermischer Widerstand
 - Spannung \Rightarrow Temperaturdifferenz

Thermischer Widerstand R_{th}

- Der Thermische Widerstand R_{th} hat die Einheit Kelvin pro Watt: K/W

Zahlenbeispiel

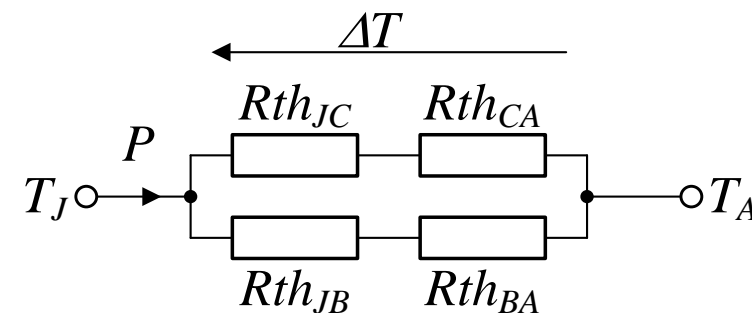
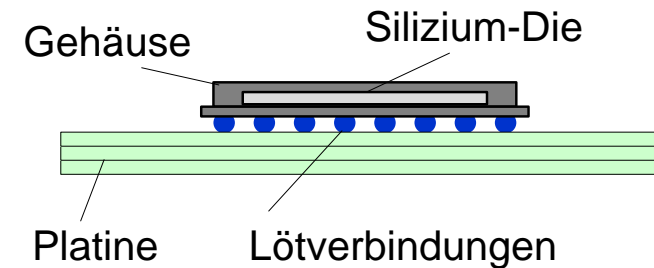
- Thermischer Widerstand $R_{th} = 20 \text{ K/W}$
- Verlustleistung $P = 1 \text{ W}$
- Umgebungstemperatur $T_A = 20^\circ\text{C}$
$$T = 20^\circ\text{C} + 1\text{W} \cdot 20\text{K/W} = 40^\circ\text{C}$$
- Verlustleistung steigt auf 2 W
$$T = 20^\circ\text{C} + 2\text{W} \cdot 20\text{K/W} = 60^\circ\text{C}$$
- Zusätzlich steigt Umgebungstemperatur auf 30°C
$$T = 30^\circ\text{C} + 2\text{W} \cdot 20\text{K/W} = 70^\circ\text{C}$$

Rechnen mit Thermischen Widerständen

- Bei einer „Reihenschaltung“ mehrerer Übergänge werden die thermischen Widerstände addiert
- Auch eine Parallelschaltung ist möglich

Beispiel: IC auf Platine

- Bild zeigt Motherboard eines PCs
- Verlustleistung entsteht in Silizium-Die (Halbleiterplättchen) innerhalb des Gehäuses
- Wärmeabfuhr über Oberfläche des Gehäuses oder über Platine möglich
- Zwei parallele Pfade mit je zwei thermischen Widerständen
 - J = Junction (Silizium-Die)
 - C = Case (Gehäuse)
 - B = Board (Platine)
 - A = Ambient (Umgebung)

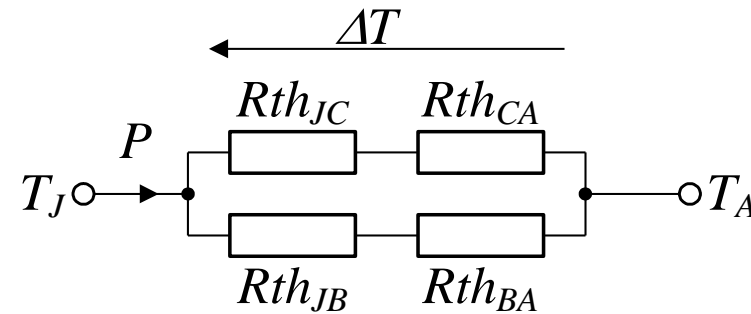


Rechnen mit Thermischen Widerständen (II)

- Oft ist ein thermischer Pfad dominant
- Dann nur Betrachtung dieses Pfades

Zahlenbeispiel aus Literatur „Maniktala“

- Hauptwärmeabfuhr über Platine
 - Pfad über Gehäuse vernachlässigt
- Wärmewiderstand Junction-Board relativ gering
 - Hauptwiderstand ist Board-Ambient
- Werte für Platinen
 - 4-lagige Platine: ~ 25 K/W
 - 2-lagige Platine: ~ 45 K/W
- Max. Sperrschichttemperatur (junction temperature) auf dem Silizium-Die theoretisch etwa 180°C
 - Sicherheitshalber nur 130°C bis 160°C
- Bis zu welcher Leistung reicht Kühlung über Platine aus?



$$\Delta T = 100 \sim 140 \text{ K, 4-lagige Platine} \rightarrow P = \Delta T / R_{th} = 100 \text{ K} / 25 \text{ K/W} = 4 \text{ W}$$

Kühlung komplexer Bausteine

- Für moderne CPUs kann Verlustleistung bis 100 W betragen
 - GPUs auf Grafikkarten sogar noch mehr
- Zahlenwerte für „4th Generation Intel® Core™ Processor Family“, z.B. i7, i5, i3
<http://www.intel.com/content/dam/www/public/us/en/documents/datasheets/4th-gen-core-family-desktop-vol-1-datasheet.pdf>
- 4C/GT2 95W: Quad Core Processor, 2014, spezifiziert für 95W
 - 88 W Thermal Design Power (TDP): Maximal erlaubte Verlustleistung
 - “TDP is not the maximum power that the processor can dissipate.”
 - „The processor will stop all execution when the junction temperature exceeds approximately 130 °C.”
 - $T_{LA} = 40^{\circ}\text{C}$ Umgebungstemperatur
 - Airflow = 3100 RPM Lüftergeschwindigkeit
 - $R_{th_{CA}} = 0,358 \text{ K/W}$ Wärmewiderstand für Kühler
 - $T_{CASE_MAX} = 74^{\circ}\text{C}$ Höchsttemperatur am IC-Gehäuse
- Kontrollrechnung: $P = \Delta T / R_{th} = 34\text{K} / 0,358 \text{ K/W} = 95 \text{ W}$
- Wärmewiderstand des Gehäuses: $R_{th} = \Delta T / P = (130^{\circ}\text{C} - 74^{\circ}\text{C}) / 95 \text{ W} = 0,589 \text{ K/W}$
→ Deutlich geringer als für Kühlung über Platine