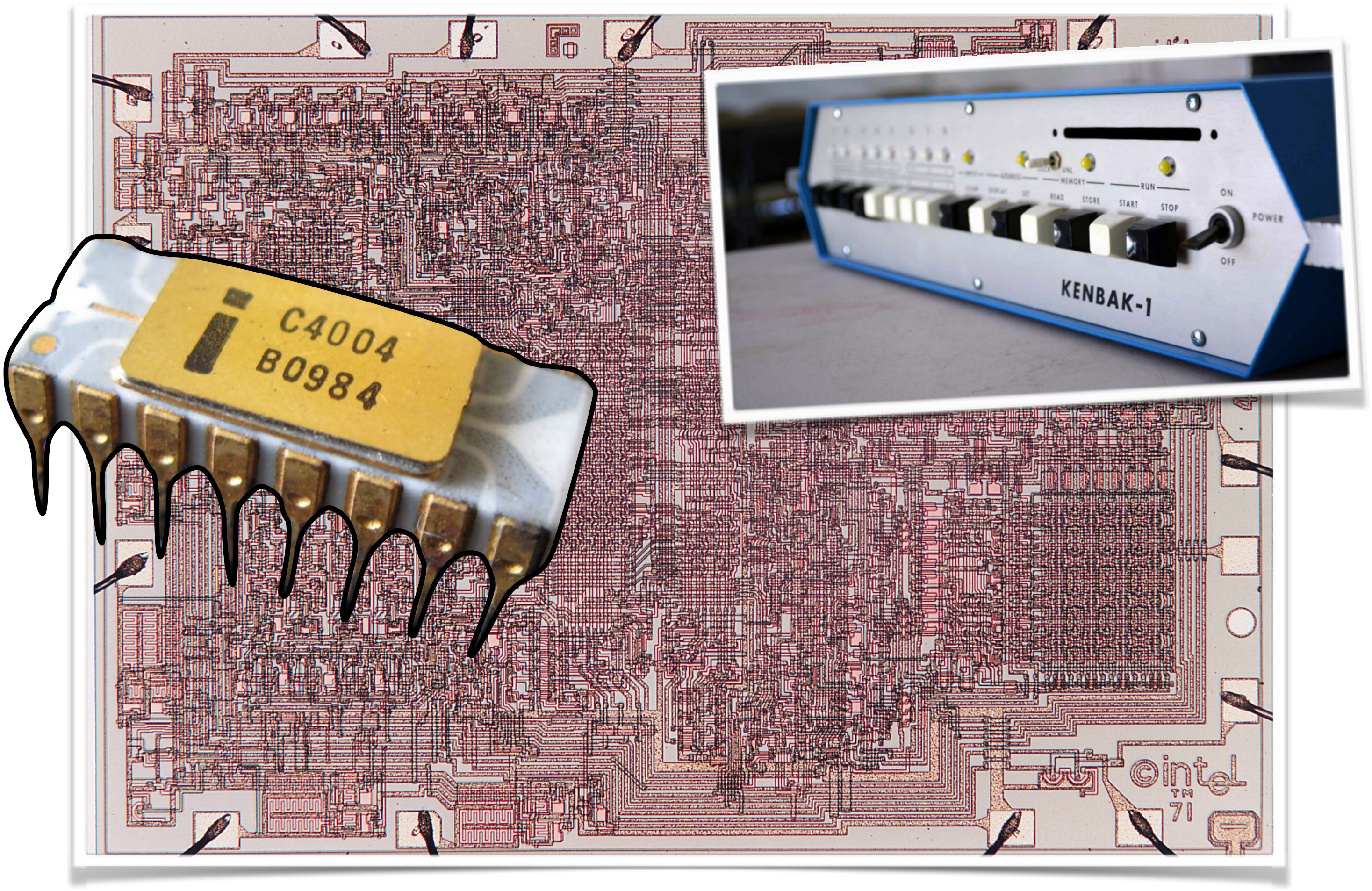
Réplique du Kenbak-1

Comment fonctionne un microcontrôleur/ microprocesseur et comment ont-t-ils révolutionné l'informatique ?

Gilles DEVILLERS / Groupe B2 / 2016 - 2017



Université François Rabelais
Institut Universitaire et Technologique de Tours
Département Génie Électronique et Informatique Industrielle

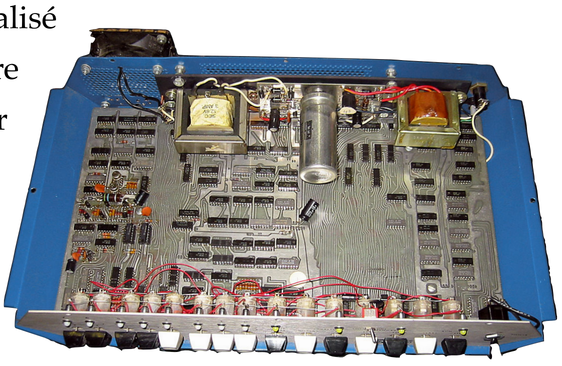
Groupe 218
Coach : Armel BRUNO

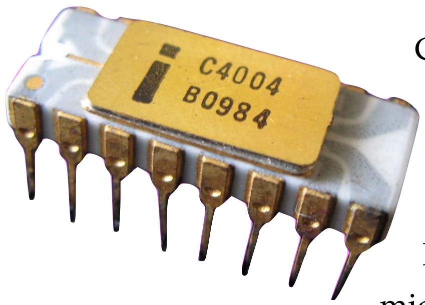
Projet tutoré en autonomie 1ère année
Véronique AUGER

Introduction

Dans le cadre de notre projet tutoré en autonomie de 1ère année, nous avons choisi de réaliser une réplique fonctionnelle du Kenbak-1 : le premier ordinateur pour particuliers disponible dans le commerce. À sa sortie en 1971, il était possible de l'acheter en magasin, de le brancher et de directement commencer à l'utiliser. C'était une très grande avancée pour l'informatique personnelle. Avant cela, l'informatique était réservée aux hobbyistes confirmés qui devaient acheter un kit, le souder et le débogguer eux-mêmes.

Une des particularités du Kenbak-1, réalisé uniquement par John Blankenbaker, était d'être dépourvu de microprocesseur (ou de microcontrôleur de façon plus générale). Toute la logique qui permettait à cet ordinateur de fonctionner était réalisée et câblée « à la main ». Ce qui le rendait encombrant, lourd, cher, plus long et difficile à produire, plus vulnérable aux pannes et plus difficile à diagnostiquer et à réparer.





C'est la même année que le premier microprocesseur a été introduit : l'Intel 4004. Ce petit circuit intégré contient toute la logique calculatoire nécessaire à un ordinateur pour exécuter des programmes.
Plus tard, le TMS1000 et l'Intel 8008 furent les premiers microcontrôleurs, regroupant cette fois-ci, en plus de la logique, de la mémoire, une horloge et d'autres périphériques les rendant complètement autonomes.

Cette recherche documentaire va donc apporter des éléments de réponse à la problématique suivante : comment fonctionne un microcontrôleur et comment ont-ils révolutionné l'informatique ?

Bibliographie

1. **Stratify Labs**. *How Microcontrollers Work*, 14 octobre 2013. (page consultée le 26 mars 2017) <https://stratifylabs.co/embedded%20design%20tips/2013/10/14/Tips-How-Microcontrollers-Work/>

2. **In One Lesson**. *How a CPU Works*, 15 mars 2013. (vidéo consultée le 26 mars 2017) <https://youtu.be/cNN_tTXABUA>

3. **Atmel Corporation**. *AVR Microcontrollers for High-Performance and Power-Effcient 8-bit Processing*, 2013, 6 p.

4. **Milan Verle**. *Architecture and programming of 8051 MCUs*. MikroElectronika, 2007.

5. **Kavita Char**. *Internet of Things System Design with Integrated Wireless MCUs*. Silicon Labs, 6 p.

6. **I Programmer**. *Intel — The Microprocessor Revolution*, 5 août 2010. (page consultée le 29 mars 2017) <http://www.i-programmer.info/history/machines/734-intel.html>

Mots-clés

- Informatique
- Microcontrôleur
- Microprocesseur
- Logique
- Combinatoire
- Séquentiel
- Mémoire
- Stockage
- Architecture
- Harvard
- Von Neumann
- RISC
- Composants
- Transistors
- Révolution
- Connecté
- Communication
- Interface
- Intelligent
- Optimisation
- Encombrement
- Circuits intégrés

Annexe 1 | How Microcontrollers Work

Identification

- Adresse de l'article : *https://stratifylabs.co/embedded%20design%20tips/2013/10/14/Tips-How-Microcontrollers-Work/*
- Nom du site : Stratify Labs
- Date de publication : 14 octobre 2013
- Date de consultation : 26 mars 2017

Évaluation du contenu

Critère d'évaluation	Faible	Moyen	Bien	Très bien
Degré de fiabilité de l'informatique				X
Utilité, accessibilité, pertinence de ces informations				X
Utilité des illustrations				X
Qualité, nombre…				X

Mots clés

- Microcontrôleur
- Architecture
- Logique
- Silicium
- Transistor
- Bascule

Compte rendu

L'article est découpé en cinq parties, partant du très bas niveau et allant jusqu'au bas niveau du fonctionnement d'un **microcontrôleur**.

1. Les semi-conducteurs : les éléments physiques utilisés dans tout circuit électronique. Notamment le **silicium** et ses électrons libres qui permettent à un courant électrique de circuler dans ce matériau.
2. Les transistors : comment sont assemblés les éléments physiques afin de réaliser des « **jonctions** ». Dans le cas du transistor, ces jonctions font office d'interrupteur contrôlé par un courant électrique plutôt que par un Homme.

3. <u>Portes logiques</u> : en assemblant les transistors de différentes manières, on obtient des éléments appelées portes logiques qui laissent passer le courant électrique sous certaines **conditions**.

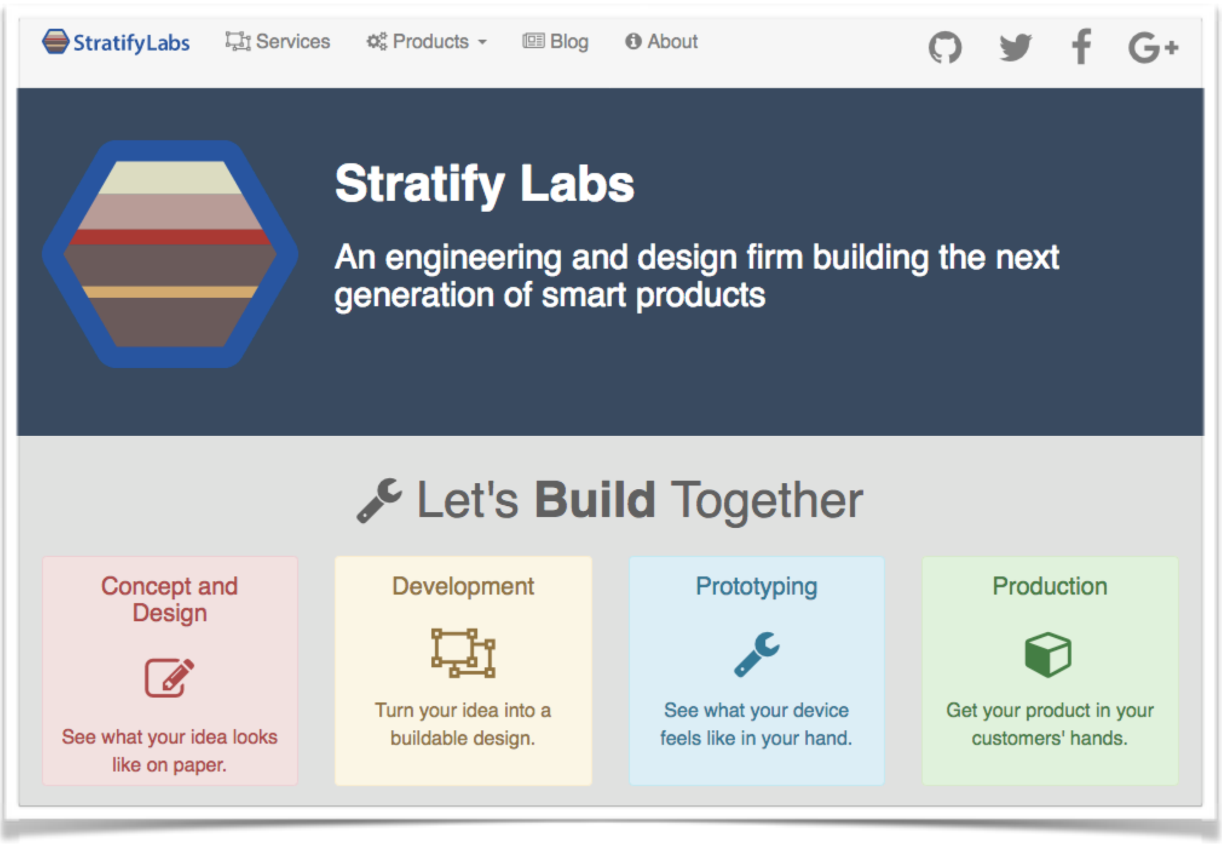
4. <u>Logique combinatoire et séquentielle</u> : ces portes logiques sont elles aussi assemblées de différentes façons afin d'obtenir des systèmes capables d'**additionner** des nombres, de les **mémoriser**, etc.

5. <u>L'architecture</u> : tous ces circuits logiques individuels (mémoire, processeur…) **communiquent** d'une certaine façon qui est propre à chaque famille de microcontrôleur.

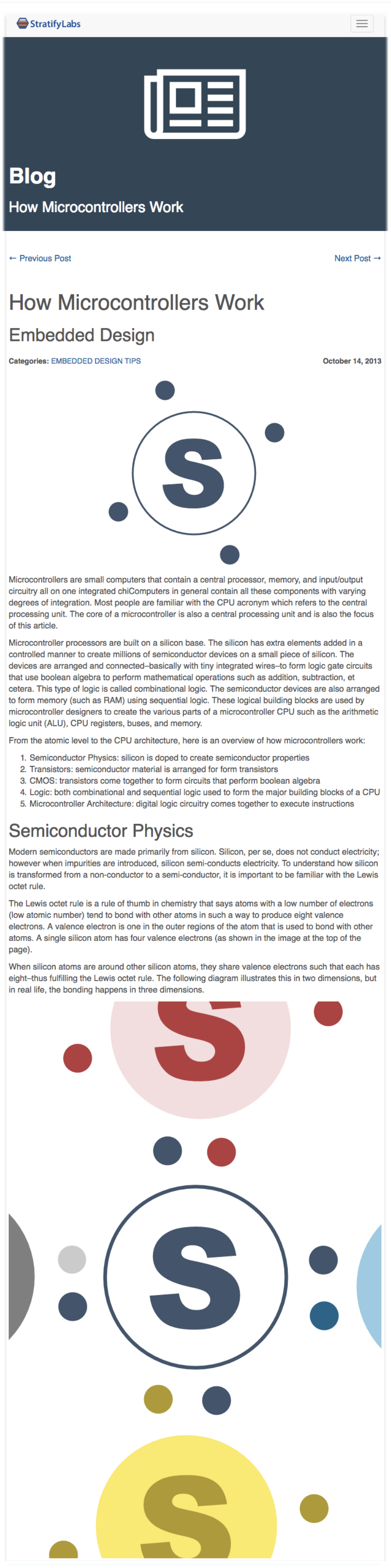
Avis

Cet article est un plaisir à lire. Il suit une progression très logique, partant du très bas niveau jusqu'au haut niveau en expliquant simplement et efficacement les différentes parties et comment elles interviennent les unes avec les autres. L'autre rentre juste assez dans les détails pour rester intéressant mais pas trop pour ne pas devenir ennuyeux. Les schémas, très simples, sont d'une grande aide à la compréhension des concepts présentés.

J'attribue une note de 10/10 à cet article pour sa clarté et sa cohérence avec le sujet de cette recherche documentaire.


StratifyLabs
Services
Products
Blog
About
Stratify Labs
An engineering and design firm building the next generation of smart products
Let's Build Together
Concept and Design
See what your idea looks like on paper.
Development
Turn your idea into a buildable design.
Prototyping
See what your device feels like in your hand.
Production
Get your product in your customers' hands.

Page d'accueil du site


StratifyLabs

Blog

How Microcontrollers Work

← Previous Post   Next Post →

How Microcontrollers Work

Embedded Design

Categories: EMBEDDED DESIGN TIPS   October 14, 2013

Microcontrollers are small computers that contain a central processor, memory, and input/output circuitry all on one integrated chiComputers in general contain all these components with varying degrees of integration. Most people are familiar with the CPU acronym which refers to the central processing unit. The core of a microcontroller is also a central processing unit and is also the focus of this article.

Microcontroller processors are built on a silicon base. The silicon has extra elements added in a controlled manner to create millions of semiconductor devices on a small piece of silicon. The devices are arranged and connected–basically with tiny integrated wires–to form logic gate circuits that use boolean algebra to perform mathematical operations such as addition, subtraction, et cetera. This type of logic is called combinational logic. The semiconductor devices are also arranged to form memory (such as RAM) using sequential logic. These logical building blocks are used by microcontroller designers to create the various parts of a microcontroller CPU such as the arithmetic logic unit (ALU), CPU registers, buses, and memory.

From the atomic level to the CPU architecture, here is an overview of how microcontrollers work:

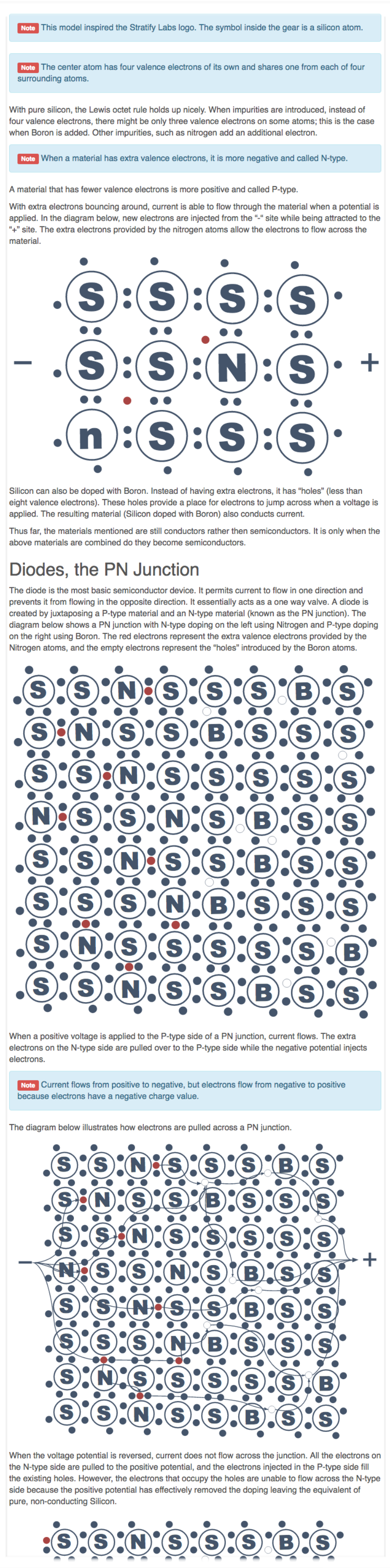
1. Semiconductor Physics: silicon is doped to create semiconductor properties
2. Transistors: semiconductor material is arranged for form transistors
3. CMOS: transistors come together to form circuits that perform boolean algebra
4. Logic: both combinational and sequential logic used to form the major building blocks of a CPU
5. Microcontroller Architecture: digital logic circuitry comes together to execute instructions

Semiconductor Physics

Modern semiconductors are made primarily from silicon. Silicon, per se, does not conduct electricity; however when impurities are introduced, silicon semi-conducts electricity. To understand how silicon is transformed from a non-conductor to a semi-conductor, it is important to be familiar with the Lewis octet rule.

The Lewis octet rule is a rule of thumb in chemistry that says atoms with a low number of electrons (low atomic number) tend to bond with other atoms in such a way to produce eight valence electrons. A valence electron is one in the outer regions of the atom that is used to bond with other atoms. A single silicon atom has four valence electrons (as shown in the image at the top of the page).

When silicon atoms are around other silicon atoms, they share valence electrons such that each has eight–thus fulfilling the Lewis octet rule. The following diagram illustrates this in two dimensions, but in real life, the bonding happens in three dimensions.


Note This model inspired the Stratify Labs logo. The symbol inside the gear is a silicon atom.

Note The center atom has four valence electrons of its own and shares one from each of four surrounding atoms.

With pure silicon, the Lewis octet rule holds up nicely. When impurities are introduced, instead of four valence electrons, there might be only three valence electrons on some atoms; this is the case when Boron is added. Other impurities, such as nitrogen add an additional electron.

Note When a material has extra valence electrons, it is more negative and called N-type.

A material that has fewer valence electrons is more positive and called P-type.

With extra electrons bouncing around, current is able to flow through the material when a potential is applied. In the diagram below, new electrons are injected from the "-" site while being attracted to the "+" site. The extra electrons provided by the nitrogen atoms allow the electrons to flow across the material.

Silicon can also be doped with Boron. Instead of having extra electrons, it has "holes" (less than eight valence electrons). These holes provide a place for electrons to jump across when a voltage is applied. The resulting material (Silicon doped with Boron) also conducts current.

Thus far, the materials mentioned are still conductors rather then semiconductors. It is only when the above materials are combined do they become semiconductors.

Diodes, the PN Junction

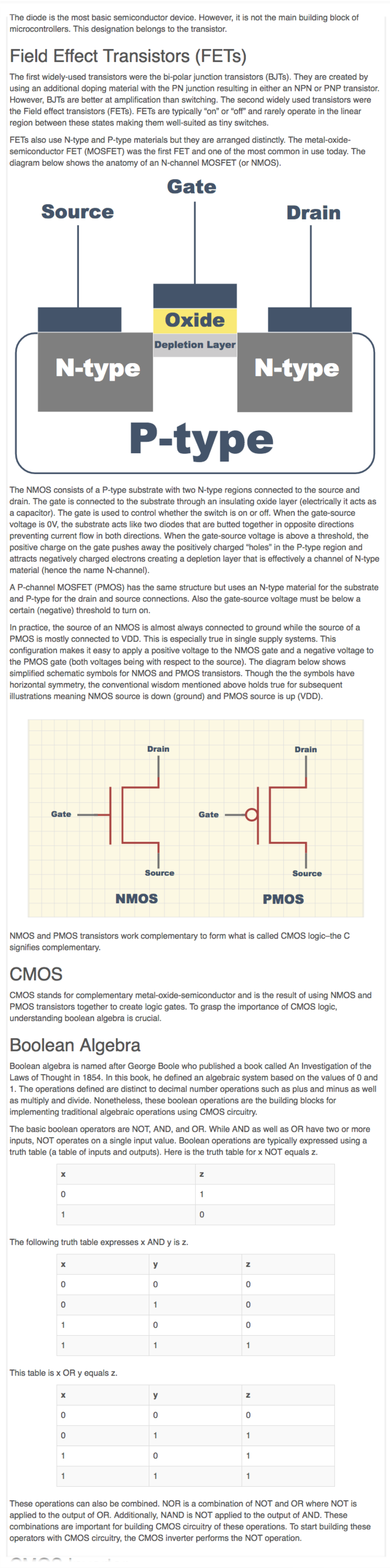
The diode is the most basic semiconductor device. It permits current to flow in one direction and prevents it from flowing in the opposite direction. It essentially acts as a one way valve. A diode is created by juxtaposing a P-type material and an N-type material (known as the PN junction). The diagram below shows a PN junction with N-type doping on the left using Nitrogen and P-type doping on the right using Boron. The red electrons represent the extra valence electrons provided by the Nitrogen atoms, and the empty electrons represent the "holes" introduced by the Boron atoms.

When a positive voltage is applied to the P-type side of a PN junction, current flows. The extra electrons on the N-type side are pulled over to the P-type side while the negative potential injects electrons.

Note Current flows from positive to negative, but electrons flow from negative to positive because electrons have a negative charge value.

The diagram below illustrates how electrons are pulled across a PN junction.

When the voltage potential is reversed, current does not flow across the junction. All the electrons on the N-type side are pulled to the positive potential, and the electrons injected in the P-type side fill the existing holes. However, the electrons that occupy the holes are unable to flow across the N-type side because the positive potential has effectively removed the doping leaving the equivalent of pure, non-conducting Silicon.


The diode is the most basic semiconductor device. However, it is not the main building block of microcontrollers. This designation belongs to the transistor.

Field Effect Transistors (FETs)

The first widely-used transistors were the bi-polar junction transistors (BJTs). They are created by using an additional doping material with the PN junction resulting in either an NPN or PNP transistor. However, BJTs are better at amplification than switching. The second widely used transistors were the Field effect transistors (FETs). FETs are typically "on" or "off" and rarely operate in the linear region between these states making them well-suited as tiny switches.

FETs also use N-type and P-type materials but they are arranged distinctly. The metal-oxide-semiconductor FET (MOSFET) was the first FET and one of the most common in use today. The diagram below shows the anatomy of an N-channel MOSFET (or NMOS).

The NMOS consists of a P-type substrate with two N-type regions connected to the source and drain. The gate is connected to the substrate through an insulating oxide layer (electrically it acts as a capacitor). The gate is used to control whether the switch is on or off. When the gate-source voltage is 0V, the substrate acts like two diodes that are butted together in opposite directions preventing current flow in both directions. When the gate-source voltage is above a threshold, the positive charge on the gate pushes away the positively charged "holes" in the P-type region and attracts negatively charged electrons creating a depletion layer that is effectively a channel of N-type material (hence the name N-channel).

A P-channel MOSFET (PMOS) has the same structure but uses an N-type material for the substrate and P-type for the drain and source connections. Also the gate-source voltage must be below a certain (negative) threshold to turn on.

In practice, the source of an NMOS is almost always connected to ground while the source of a PMOS is mostly connected to VDD. This is especially true in single supply systems. This configuration makes it easy to apply a positive voltage to the NMOS gate and a negative voltage to the PMOS gate (both voltages being with respect to the source). The diagram below shows simplified schematic symbols for NMOS and PMOS transistors. Though the the symbols have horizontal symmetry, the conventional wisdom mentioned above holds true for subsequent illustrations meaning NMOS source is down (ground) and PMOS source is up (VDD).

NMOS and PMOS transistors work complementary to form what is called CMOS logic–the C signifies complementary.

CMOS

CMOS stands for complementary metal-oxide-semiconductor and is the result of using NMOS and PMOS transistors together to create logic gates. To grasp the importance of CMOS logic, understanding boolean algebra is crucial.

Boolean Algebra

Boolean algebra is named after George Boole who published a book called An Investigation of the Laws of Thought in 1854. In this book, he defined an algebraic system based on the values of 0 and 1. The operations defined are distinct to decimal number operations such as plus and minus as well as multiply and divide. Nonetheless, these boolean operations are the building blocks for implementing traditional algebraic operations using CMOS circuitry.

The basic boolean operators are NOT, AND, and OR. While AND as well as OR have two or more inputs, NOT operates on a single input value. Boolean operations are typically expressed using a truth table (a table of inputs and outputs). Here is the truth table for x NOT equals z.

x	z
0	1
1	0

The following truth table expresses x AND y is z.

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

This table is x OR y equals z.

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

These operations can also be combined. NOR is a combination of NOT and OR where NOT is applied to the output of OR. Additionally, NAND is NOT applied to the output of AND. These combinations are important for building CMOS circuitry of these operations. To start building these operators with CMOS circuitry, the CMOS inverter performs the NOT operation.

Extrait de l'article consulté

Annexe 2 | How a CPU Works

Identification

- Adresse de la vidéo : *https://youtu.be/cNN_tTXABUA*
- Nom du site : Youtube
- Réalisateur : In One Lesson
- Date de publication : 15 mars 2013
- Date de consultation : 26 mars 2017

Évaluation du contenu

Critère d'évaluation	Faible	Moyen	Bien	Très bien
Degré de fiabilité de l'informatique			X	
Utilité, accessibilité, pertinence de ces informations			X	
Utilité des illustrations				X
Qualité, nombre…				X

Mots clés

- Microcontrôleur
- Architecture
- Unité de contrôle
- Logique
- Programmation
- Mémoire vive
- Assembleur
- Bus de données
- Instructions

Compte rendu

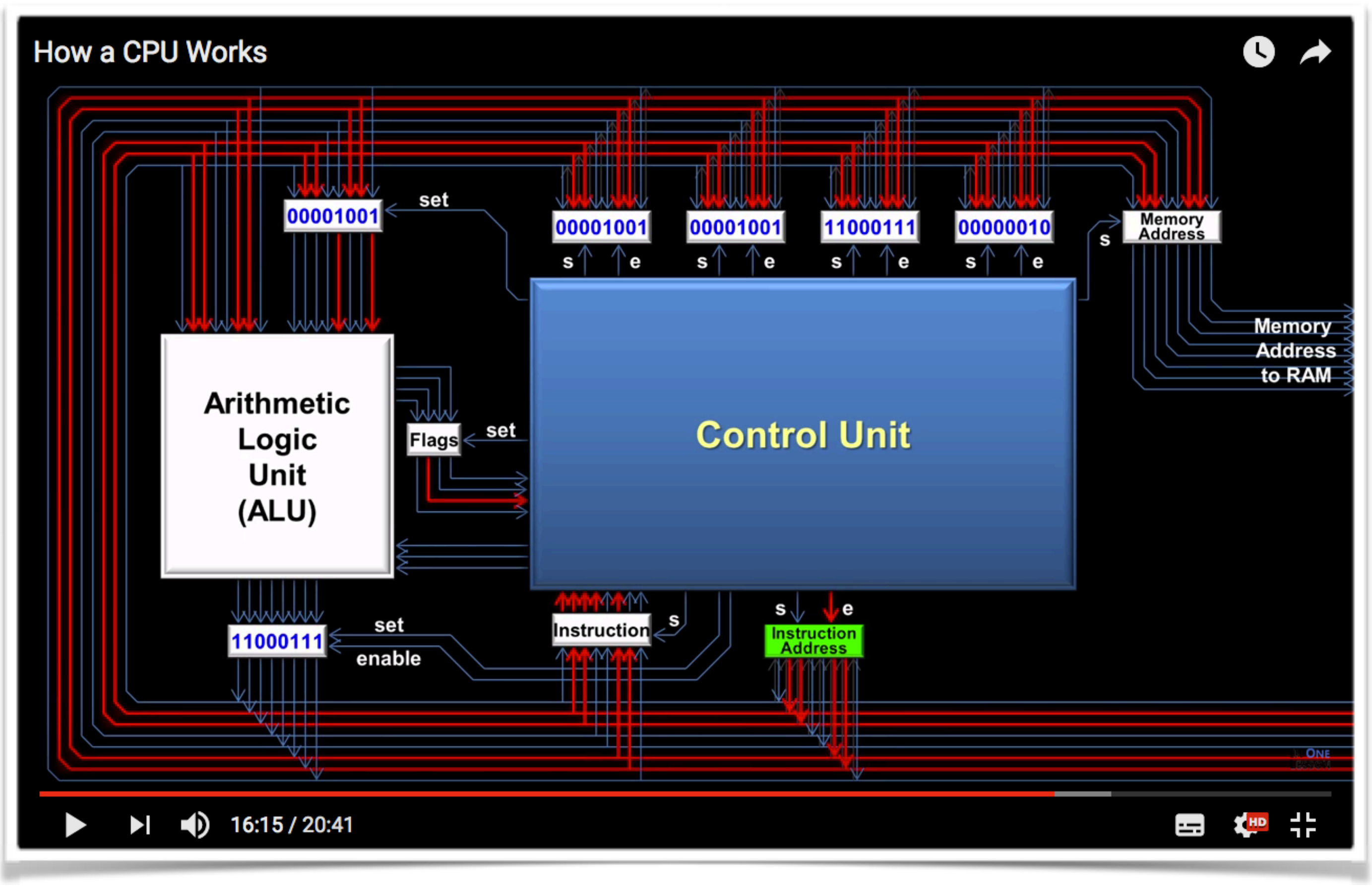
Cette vidéo, inspirée du livre « *But How Do It Know* » de John Scott, explique principalement le **fonctionnement** interne du **microprocesseur**, élément principal d'un **microcontrôleur**. Contrairement au document précédent, l'auteur part du haut niveau pour aller vers le bas niveau du fonctionnement du microprocesseur.

Chaque partie du microprocesseur est détaillée. Premièrement l'**architecture** Von Neumann et le stockage des instructions sous forme binaire dans la mémoire vive. Puis la façon dont l'**unité de contrôle** gère les opérations et les transferts de données entre l'**UAL** (Unité Arithmétique et Logique, chargée d'effectuer les calculs) et les **registres** (mémoire temporaire). Le **bus** de données principal montre l'**interface** entre le processeur et les mémoires ou périphériques externes.

Avis

La vidéo explique le fonctionnement d'un microprocesseur de façon simple. Il est intéressant de le mettre en perspective du premier article pour avoir plus de détails sur le fonctionnement global du microprocesseur et sa communication avec la mémoire vive. On comprend bien la façon de programmer un microcontrôleur et comment les programmes sont exécutés sur une architecture Von Neumann.

J'attribue à cette vidéo la note de 8/10 car bien que son accessibilité en face son intérêt, quelques détails supplémentaires pourraient être souhaités comme le fonctionnement interne de l'unité de contrôle.



Capture d'écran de la vidéo

Annexe 3 | AVR Microcontrollers for High-Performance and Power-Efficient 8-bit Processing

Identification

- Auteur : Atmel Corporation
- Titre : AVR Microcontrollers for High Performance and Power-Efficient 8-bit Processing
- Date de 1ère parution : 2013

Évaluation du contenu

Critère d'évaluation	Faible	Moyen	Bien	Très bien
Degré de fiabilité de l'informatique				X
Utilité, accessibilité, pertinence de ces informations			X	
Utilité des illustrations				X
Qualité, nombre…			X	

Mots clés

- Microcontrôleur
- Architecture
- Structure
- Logique
- Horloge
- Fréquence
- RISC
- Bus de données
- Instructions

Compte rendu

Ce document publié par Atmel Corporation en 2013 présente la gamme de **microcontrôleurs** vendue par Atmel : AVR.

Ceux-ci fonctionnent sur la base d'une **architecture Harvard** modifiée, cela signifie que la **mémoire** stockant le programme et la **mémoire** dans laquelle est située les données du programme sont séparées en deux. Il est d'usage d'utiliser deux **technologies** de stockage **différentes** dans une telle architecture, ici la **mémoire flash** et la **mémoire vive statique** respectivement.

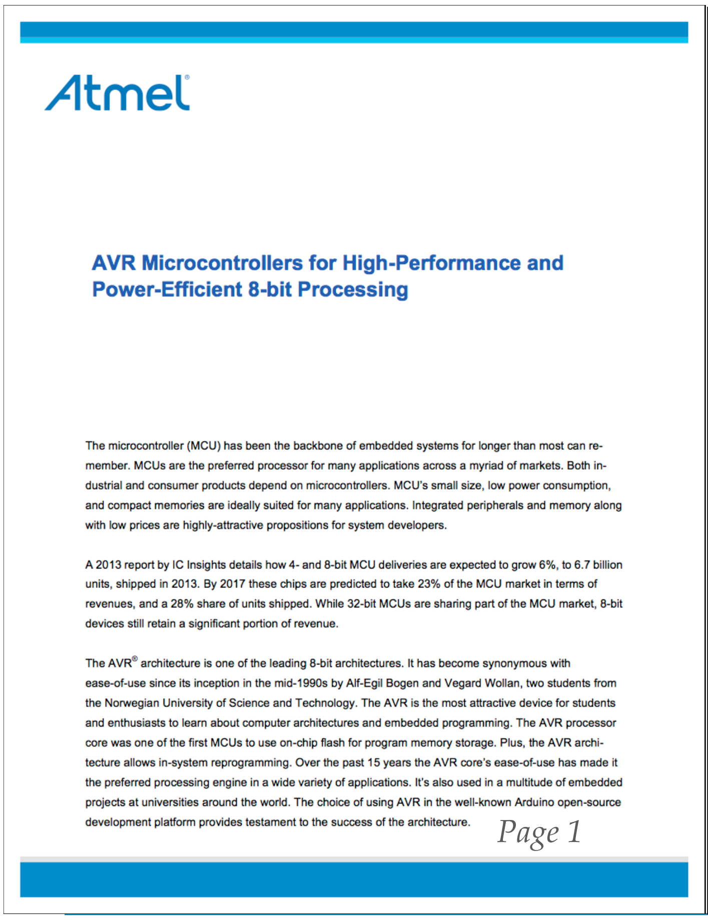
Les **microcontrôleurs** AVR ont un processeur conçu sur l'**architecture RISC** (**R**educed **I**nstruction **S**et **C**omputer : processeur à jeu d'instruction réduit), qui s'oppose à l'architecture **CISC**. Ce principe donne au processeur un nombre **important** d'instructions mais qui exécutent chacune des tâches **élémentaires**, ce qui peut **allonger** la taille du programme pour des opérations complexes mais également **accélérer** le temps d'exécution de ces plus petites instructions.

Les connexions entre les différents **périphériques internes** du **microcontrôleur** sont détaillées ainsi que les **interfaces** de sorties qui permettent à l'Homme de **communiquer** avec ce dernier.

Avis

Ce document technique de présentation est très clair et je lui attribuerai la note de 8/10. Il explique pourquoi il est intéressant de choisir les microcontrôleurs AVR mais en restant dans un cadre technique, en présentant le fonctionnement de ceux-ci et leurs différents composants internes. Les schémas sont d'une grande aide pour comprendre le fonctionnement de ce type de microcontrôleur et comment les connexions entre les périphériques internes et externes ont lieu.

Il est d'autant plus intéressant de s'intéresser aux microcontrôleurs AVR puisqu'ils sont majoritairement utilisés à l'IUT GEII. C'est également ce type de microcontrôleur qui sera présent dans notre projet pour remplacer toute la logique câblée de l'époque du Kenbak-1.



Atmel

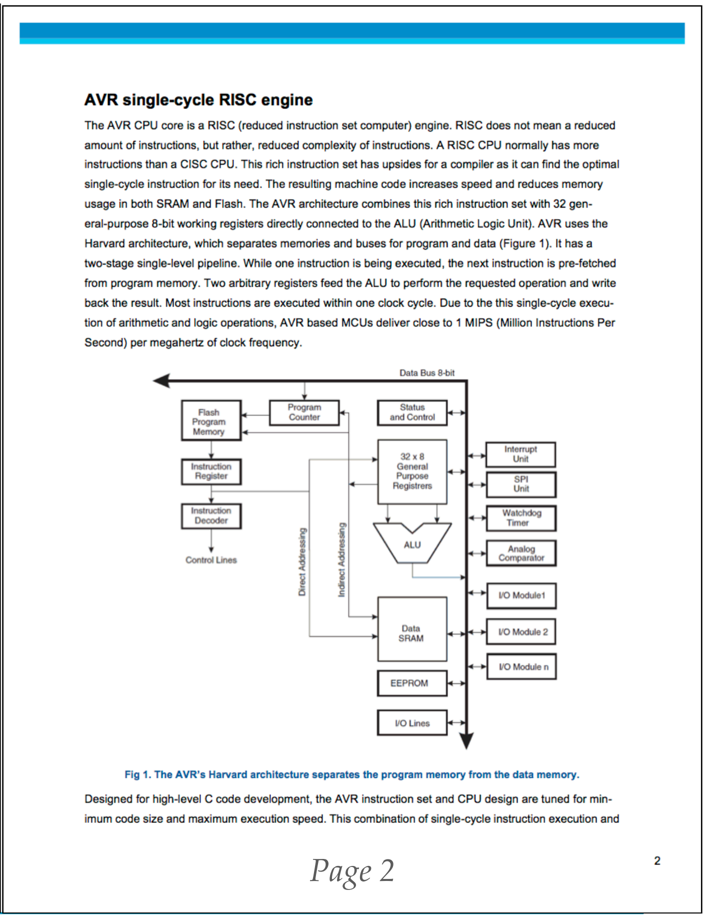
AVR Microcontrollers for High-Performance and Power-Efficient 8-bit Processing

The microcontroller (MCU) has been the backbone of embedded systems for longer than most can remember. MCUs are the preferred processor for many applications across a myriad of markets. Both industrial and consumer products depend on microcontrollers. MCU's small size, low power consumption, and compact memories are ideally suited for many applications. Integrated peripherals and memory along with low prices are highly-attractive propositions for system developers.

A 2013 report by IC Insights details how 4- and 8-bit MCU deliveries are expected to grow 6%, to 6.7 billion units, shipped in 2013. By 2017 these chips are predicted to take 23% of the MCU market in terms of revenues, and a 28% share of units shipped. While 32-bit MCUs are sharing part of the MCU market, 8-bit devices still retain a significant portion of revenue.

The AVR® architecture is one of the leading 8-bit architectures. It has become synonymous with ease-of-use since its inception in the mid-1990s by Alf-Egil Bogen and Vegard Wollan, two students from the Norwegian University of Science and Technology. The AVR is the most attractive device for students and enthusiasts to learn about computer architectures and embedded programming. The AVR processor core was one of the first MCUs to use on-chip flash for program memory storage. Plus, the AVR architecture allows in-system reprogramming. Over the past 15 years the AVR core's ease-of-use has made it the preferred processing engine in a wide variety of applications. It's also used in a multitude of embedded projects at universities around the world. The choice of using AVR in the well-known Arduino open-source development platform provides testament to the success of the architecture.

Page 1



AVR single-cycle RISC engine

The AVR CPU core is a RISC (reduced instruction set computer) engine. RISC does not mean a reduced amount of instructions, but rather, reduced complexity of instructions. A RISC CPU normally has more instructions than a CISC CPU. This rich instruction set has upsides for a compiler as it can find the optimal single-cycle instruction for its need. The resulting machine code increases speed and reduces memory usage in both SRAM and Flash. The AVR architecture combines this rich instruction set with 32 general-purpose 8-bit working registers directly connected to the ALU (Arithmetic Logic Unit). AVR uses the Harvard architecture, which separates memories and buses for program and data (Figure 1). It has a two-stage single-level pipeline. While one instruction is being executed, the next instruction is pre-fetched from program memory. Two arbitrary registers feed the ALU to perform the requested operation and write back the result. Most instructions are executed within one clock cycle. Due to the this single-cycle execution of arithmetic and logic operations, AVR based MCUs deliver close to 1 MIPS (Million Instructions Per Second) per megahertz of clock frequency.

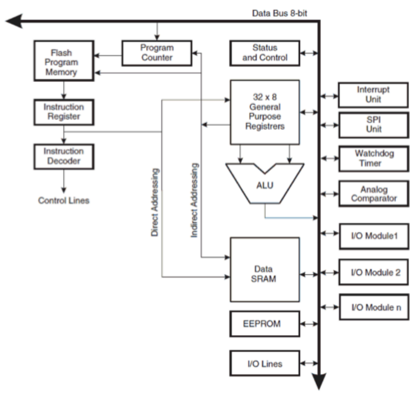
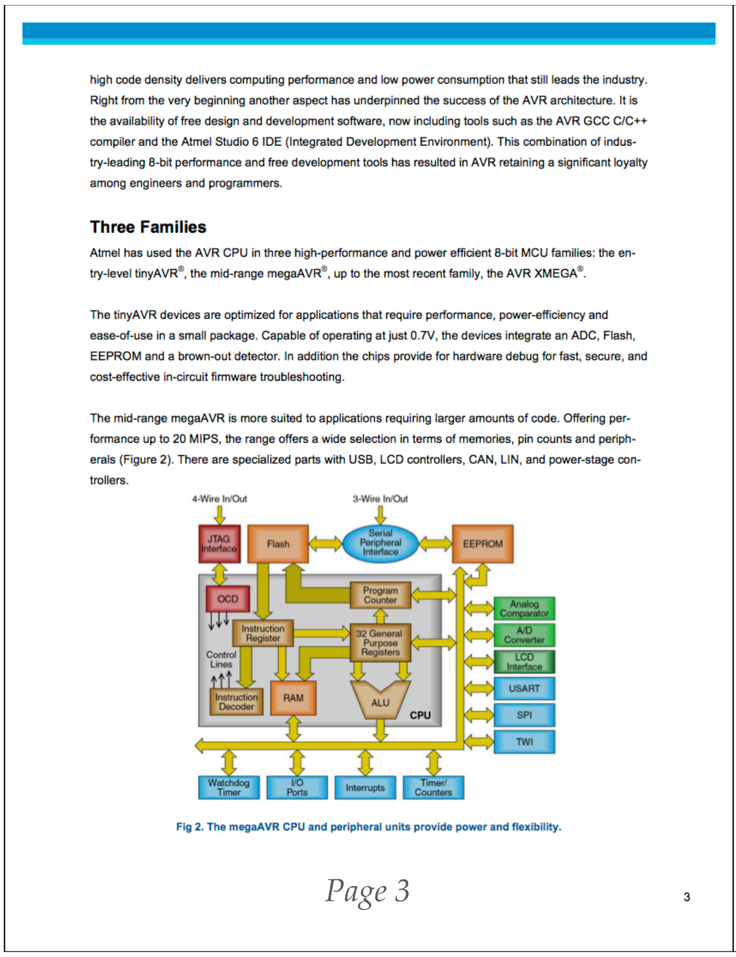


Fig 1. The AVR's Harvard architecture separates the program memory from the data memory.

Designed for high-level C code development, the AVR instruction set and CPU design are tuned for minimum code size and maximum execution speed. This combination of single-cycle instruction execution and

Page 2

2



high code density delivers computing performance and low power consumption that still leads the industry. Right from the very beginning another aspect has underpinned the success of the AVR architecture. It is the availability of free design and development software, now including tools such as the AVR GCC C/C++ compiler and the Atmel Studio 6 IDE (Integrated Development Environment). This combination of industry-leading 8-bit performance and free development tools has resulted in AVR retaining a significant loyalty among engineers and programmers.

Three Families

Atmel has used the AVR CPU in three high-performance and power efficient 8-bit MCU families: the entry-level tinyAVR®, the mid-range megaAVR®, up to the most recent family, the AVR XMEGA®.

The tinyAVR devices are optimized for applications that require performance, power-efficiency and ease-of-use in a small package. Capable of operating at just 0.7V, the devices integrate an ADC, Flash, EEPROM and a brown-out detector. In addition the chips provide for hardware debug for fast, secure, and cost-effective in-circuit firmware troubleshooting.

The mid-range megaAVR is more suited to applications requiring larger amounts of code. Offering performance up to 20 MIPS, the range offers a wide selection in terms of memories, pin counts and peripherals (Figure 2). There are specialized parts with USB, LCD controllers, CAN, LIN, and power-stage controllers.

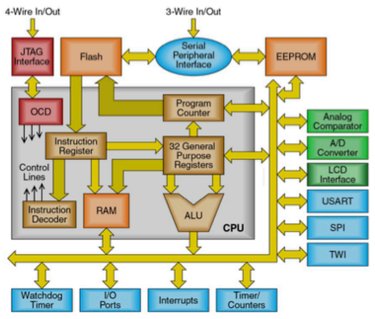
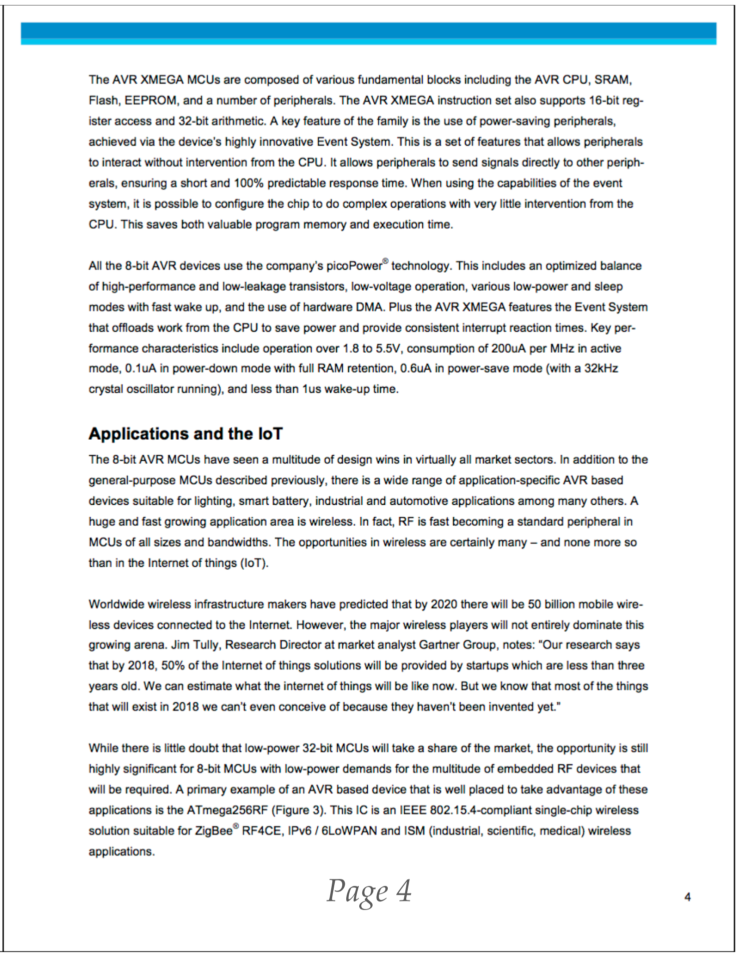


Fig 2. The megaAVR CPU and peripheral units provide power and flexibility.

Page 3

3



The AVR XMEGA MCUs are composed of various fundamental blocks including the AVR CPU, SRAM, Flash, EEPROM, and a number of peripherals. The AVR XMEGA instruction set also supports 16-bit register access and 32-bit arithmetic. A key feature of the family is the use of power-saving peripherals, achieved via the device's highly innovative Event System. This is a set of features that allows peripherals to interact without intervention from the CPU. It allows peripherals to send signals directly to other peripherals, ensuring a short and 100% predictable response time. When using the capabilities of the event system, it is possible to configure the chip to do complex operations with very little intervention from the CPU. This saves both valuable program memory and execution time.

All the 8-bit AVR devices use the company's picoPower® technology. This includes an optimized balance of high-performance and low-leakage transistors, low-voltage operation, various low-power and sleep modes with fast wake up, and the use of hardware DMA. Plus the AVR XMEGA features the Event System that offloads work from the CPU to save power and provide consistent interrupt reaction times. Key performance characteristics include operation over 1.8 to 5.5V, consumption of 200uA per MHz in active mode, 0.1uA in power-down mode with full RAM retention, 0.6uA in power-save mode (with a 32kHz crystal oscillator running), and less than 1us wake-up time.

Applications and the IoT

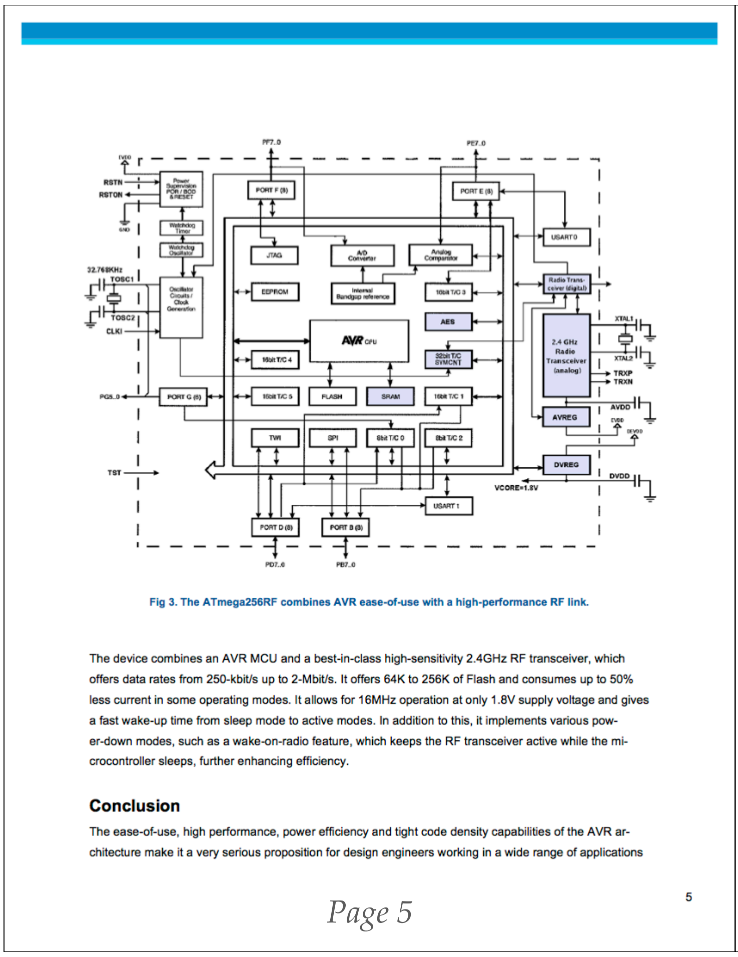
The 8-bit AVR MCUs have seen a multitude of design wins in virtually all market sectors. In addition to the general-purpose MCUs described previously, there is a wide range of application-specific AVR based devices suitable for lighting, smart battery, industrial and automotive applications among many others. A huge and fast growing application area is wireless. In fact, RF is fast becoming a standard peripheral in MCUs of all sizes and bandwidths. The opportunities in wireless are certainly many – and none more so than in the Internet of things (IoT).

Worldwide wireless infrastructure makers have predicted that by 2020 there will be 50 billion mobile wireless devices connected to the Internet. However, the major wireless players will not entirely dominate this growing arena. Jim Tully, Research Director at market analyst Gartner Group, notes: "Our research says that by 2018, 50% of the Internet of things solutions will be provided by startups which are less than three years old. We can estimate what the internet of things will be like now. But we know that most of the things that will exist in 2018 we can't even conceive of because they haven't been invented yet."

While there is little doubt that low-power 32-bit MCUs will take a share of the market, the opportunity is still highly significant for 8-bit MCUs with low-power demands for the multitude of embedded RF devices that will be required. A primary example of an AVR based device that is well placed to take advantage of these applications is the ATmega256RF (Figure 3). This IC is an IEEE 802.15.4-compliant single-chip wireless solution suitable for ZigBee® RF4CE, IPv6 / 6LoWPAN and ISM (industrial, scientific, medical) wireless applications.

Page 4

4



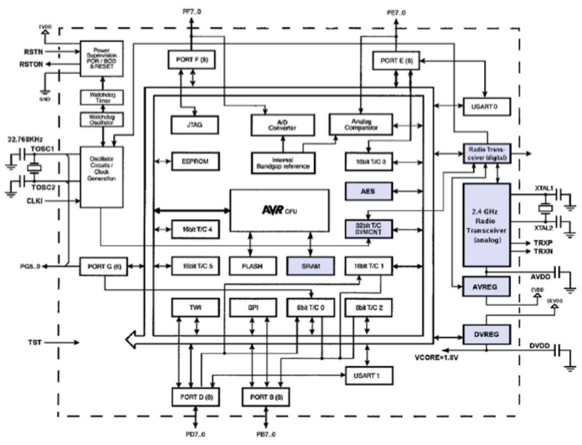


Fig 3. The ATmega256RF combines AVR ease-of-use with a high-performance RF link.

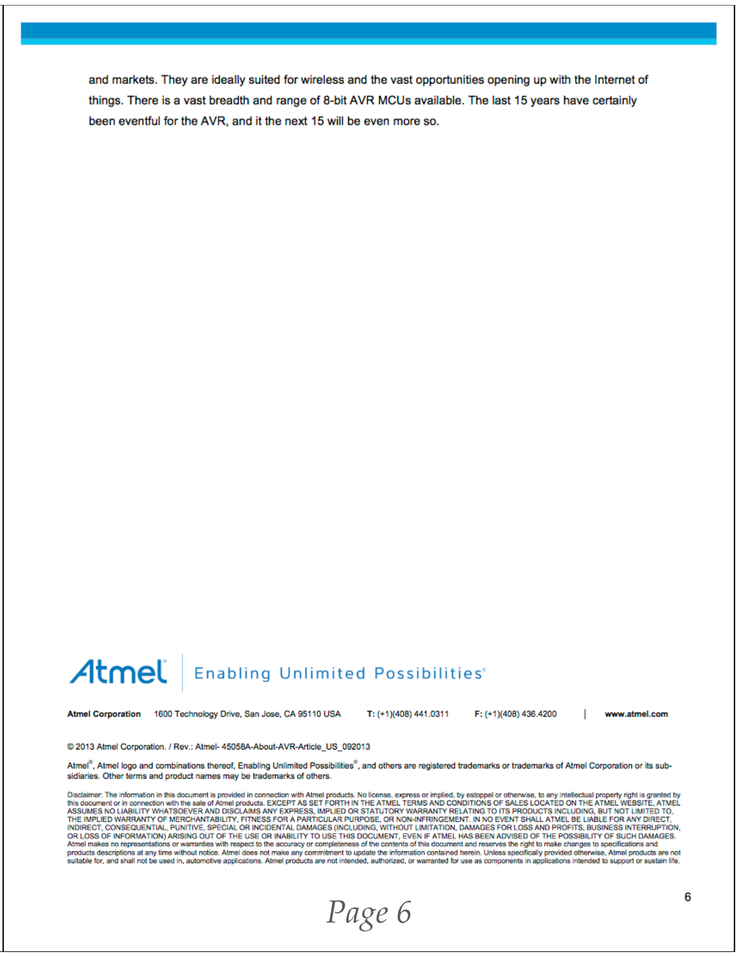
The device combines an AVR MCU and a best-in-class high-sensitivity 2.4GHz RF transceiver, which offers data rates from 250-kbit/s up to 2-Mbit/s. It offers 64K to 256K of Flash and consumes up to 50% less current in some operating modes. It allows for 16MHz operation at only 1.8V supply voltage and gives a fast wake-up time from sleep mode to active modes. In addition to this, it implements various power-down modes, such as a wake-on-radio feature, which keeps the RF transceiver active while the microcontroller sleeps, further enhancing efficiency.

Conclusion

The ease-of-use, high performance, power efficiency and tight code density capabilities of the AVR architecture make it a very serious proposition for design engineers working in a wide range of applications

5

Page 5



and markets. They are ideally suited for wireless and the vast opportunities opening up with the Internet of things. There is a vast breadth and range of 8-bit AVR MCUs available. The last 15 years have certainly been eventful for the AVR, and it the next 15 will be even more so.

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA T: (+1)(408) 441.0311 F: (+1)(408) 436.4200 | **www.atmel.com**

6

Page 6

Annexe 4 | Architecture and programming of 8051 MCUs

Identification

- Auteur : Milan Verle
- Titre : Architecture and programming of 8051 MCUs
- Éditeur : MikroElektronika
- Date de 1ère parution : 1er janvier 2007

Évaluation du contenu

Critère d'évaluation	Faible	Moyen	Bien	Très bien
Degré de fiabilité de l'informatique			X	
Utilité, accessibilité, pertinence de ces informations				X
Utilité des illustrations				X
Qualité, nombre…			X	

Mots clés

- Microcontrôleur
- Architecture
- Unité de contrôle
- RISC
- Harvard
- Timers
- Programme
- Watchdog
- Mémoire

Compte rendu

Ce livre est une introduction à l'**architecture** et la **programmation** des microcontrôleurs 8051 introduits par Intel en 1980 et encore utilisés comme base aujourd'hui. Ceux-ci sont basés sur une architecture **Harvard** (mémoires programme et données séparées) et **CISC** (**C**omplex **I**nstruction **S**et **C**omputer : microprocesseur à jeu d'instruction étendu).

Le premier chapitre explique assez généralement ce qu'est un **microcontrôleur**, puis sa mise en marche et son fonctionnement **cyclique**.

Le deuxième chapitre se concentre sur les différents éléments indépendants qui composent le microcontrôleur, tels que :

- les <u>mémoires</u> : ROM (lecture seule), RAM (accès aléatoire), EEPROM (effaçable électriquement et programmable).
- les <u>registres</u> : mémoire **temporaire rapide** d'accès pour stocker les opérandes et résultats des calculs exécutés par le processeur.
- le <u>pointeur d'instruction</u> : un nombre qui définit la **prochaine** instruction qui va être exécutée par le processeur.
- le <u>processeur</u> : composé du **décodeur** d'instructions, de l'UAL (**U**nité **A**rithmétique et **L**ogique) et de l'**accumulateur**. Il s'occupe de l'**exécution** du programme et de la **communication** avec les autres composants.
- l'<u>oscillateur</u> : l'horloge interne synchrone qui **régit** la vitesse de fonctionnement de tout le microcontrôleur.
- les <u>timers</u> : des éléments utiles pour compter des évènements ou exécuter un programme à **intervalles** de temps **réguliers**.
- le <u>port série</u> : moyen de **communiquer** entre plusieurs microcontrôleurs.

Avis

Je noterai ce document 7/10 pour la simplicité des explications des différents termes techniques. L'assemblage des composant présents dans chaque microcontrôleur est explicité en détail. Cela aide à comprendre le fonctionnement général des différents sous-éléments et la façon dont il est possible de les utiliser sans pour autant rentrer dans le code informatique ésotérique. Plus de détails seraient pourtant intéressants sur certains périphériques présentés.

ARHITEKTURA I PROGRAMIRANJE
8051
MIKROKONTROLERA
ŠTA SU MIKROKONTROLERI I KAKO RADE?
U knjizi je detaljno objašnjeno sve što je potrebno da znate da biste uspešno ukrotili snagu modernih mikrokontrolera.
Milan Verle
MIKROKONTROLERI SERIJE 8051
Saznajte sve o "standardu 8051" i moći ćete da radite sa nekoliko stotina različitih mikrokontrolera!
KAKO IH PRIMENITI U PRAKSI?
Svi primeri su detaljno objašnjeni i ilustrovani.
Ako ste početnik u ovoj oblasti, iskoristite iskustva eksperata.
SADRŽI CD
MikroElektronika

TABLE OF CONTENTS

Annexe 5 I Internet of Things System Design with Integrated Wireless MCUs

Identification

- Auteur : Kavita Char
- Titre : Internet of Things System Design with Integrated Wireless MCUs
- Éditeur : Silicon Labs

Évaluation du contenu

Critère d'évaluation	Faible	Moyen	Bien	Très bien
Degré de fiabilité de l'informatique				X
Utilité, accessibilité, pertinence de ces informations				X
Utilité des illustrations			X	
Qualité, nombre…			X	

Mots clés

- Connexion
- Internet
- Réseaux
- Interaction
- Futur
- Interface
- Communication
- Sans fil
- Intelligent

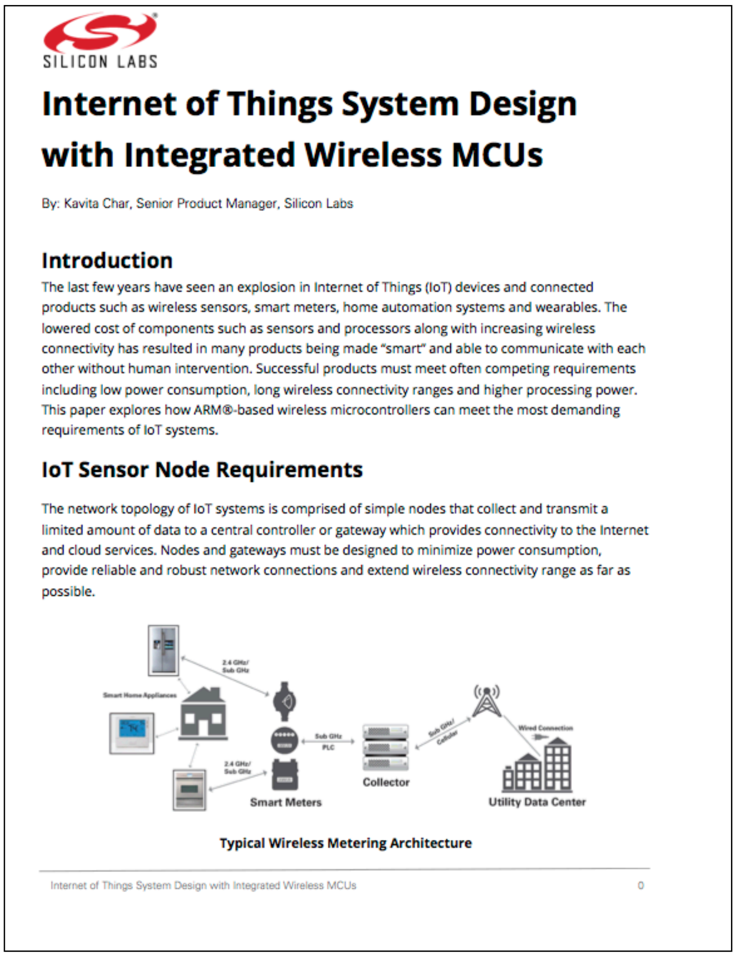
Compte rendu

Ce document explique une des utilisations des microcontrôleurs qui a **explosé** depuis quelques années : l'IoT (**I**nternet **o**f **T**hings : Internet des Objets). Ce domaine utilise des microcontrôleurs dotés de **connexions** à **internet**, souvent sans fils, afin de connecter un objet, tel qu'un réfrigérateur, une montre ou même un **casier scolaire** au monde entier.

Il est décrit la façon dont ces objets sont connectés au **réseau**, ainsi que le principe de fonctionnement de base d'un **IoT** (capteurs et données sont analysées par le microcontrôleur). La façon de choisir un microcontrôleur et une technologie sans fil est détaillée. Enfin, sont données des **règles** de conception, notamment la vitesse de **développement** et la basse **consommation** nécessaire du système.

Avis

Je noterai ce document 7/10. Il montre en effet que même après son invention qui remonte à quelques décennies, le microcontrôleur est encore au cœur de la révolution technologique actuelle. Le document s'intéresse aux bons points en matière de choix à réaliser pour l'application voulue. Des exemples et illustrations supplémentaires comme des schémas pourraient être d'une réelle aide à la compréhension de l'écosystème des objets connectés.



SILICON LABS

Internet of Things System Design with Integrated Wireless MCUs

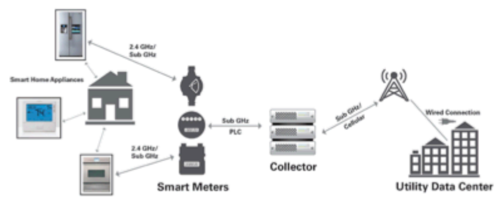
By: Kavita Char, Senior Product Manager, Silicon Labs

Introduction

The last few years have seen an explosion in Internet of Things (IoT) devices and connected products such as wireless sensors, smart meters, home automation systems and wearables. The lowered cost of components such as sensors and processors along with increasing wireless connectivity has resulted in many products being made "smart" and able to communicate with each other without human intervention. Successful products must meet often competing requirements including low power consumption, long wireless connectivity ranges and higher processing power. This paper explores how ARM®-based wireless microcontrollers can meet the most demanding requirements of IoT systems.

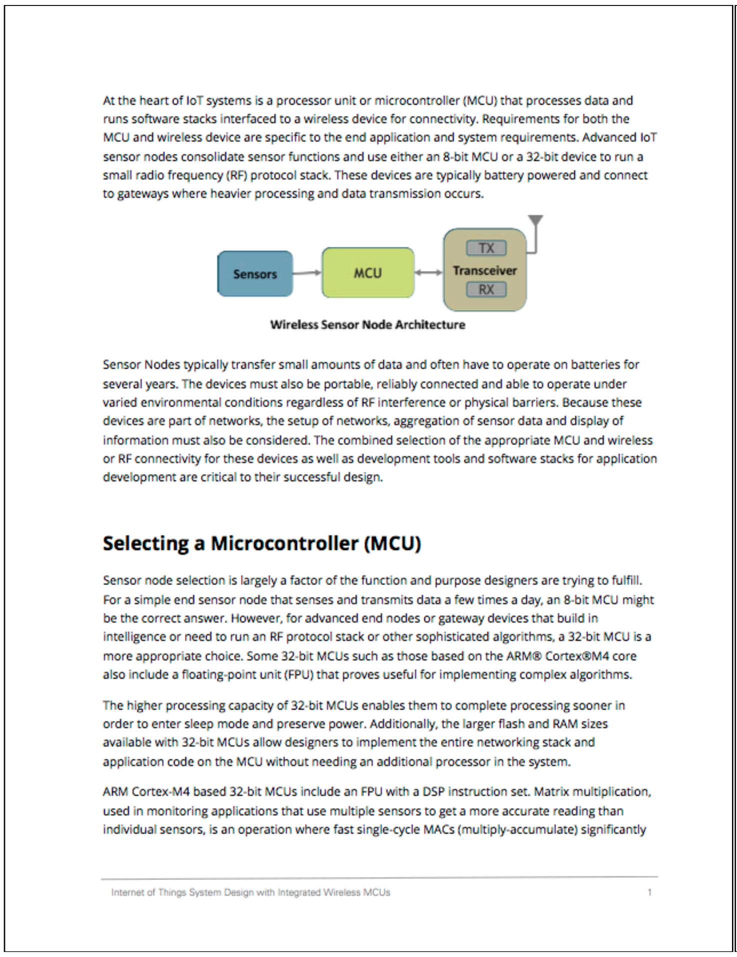
IoT Sensor Node Requirements

The network topology of IoT systems is comprised of simple nodes that collect and transmit a limited amount of data to a central controller or gateway which provides connectivity to the Internet and cloud services. Nodes and gateways must be designed to minimize power consumption, provide reliable and robust network connections and extend wireless connectivity range as far as possible.

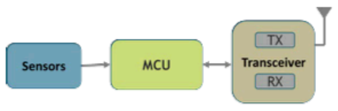


Typical Wireless Metering Architecture

Internet of Things System Design with Integrated Wireless MCUs 0



At the heart of IoT systems is a processor unit or microcontroller (MCU) that processes data and runs software stacks interfaced to a wireless device for connectivity. Requirements for both the MCU and wireless device are specific to the end application and system requirements. Advanced IoT sensor nodes consolidate sensor functions and use either an 8-bit MCU or a 32-bit device to run a small radio frequency (RF) protocol stack. These devices are typically battery powered and connect to gateways where heavier processing and data transmission occurs.



Wireless Sensor Node Architecture

Sensor Nodes typically transfer small amounts of data and often have to operate on batteries for several years. The devices must also be portable, reliably connected and able to operate under varied environmental conditions regardless of RF interference or physical barriers. Because these devices are part of networks, the setup of networks, aggregation of sensor data and display of information must also be considered. The combined selection of the appropriate MCU and wireless or RF connectivity for these devices as well as development tools and software stacks for application development are critical to their successful design.

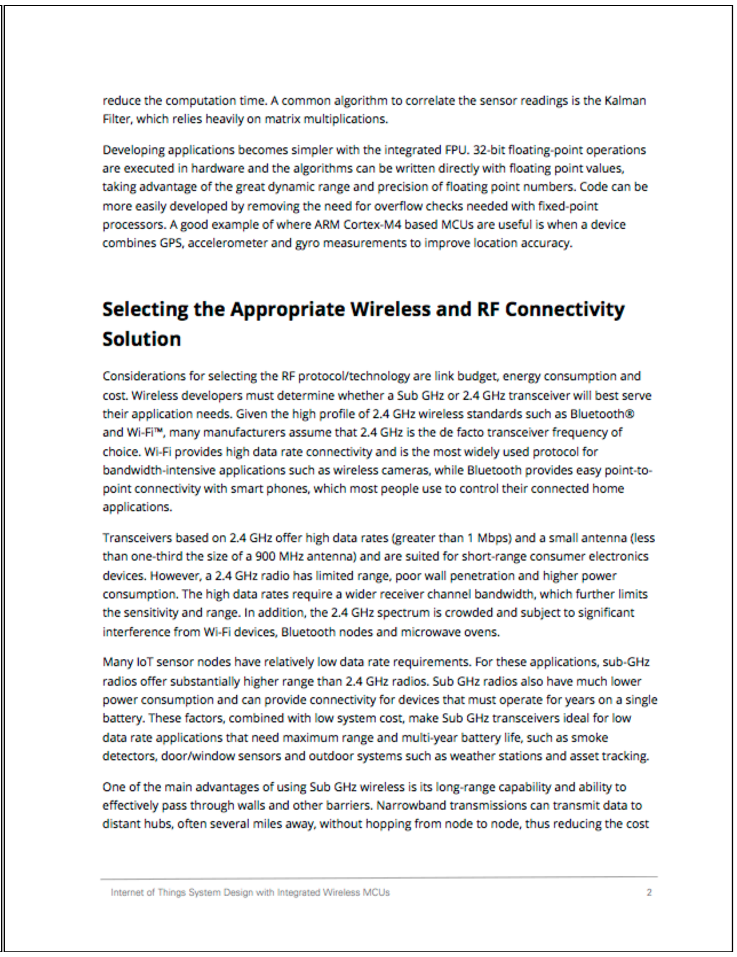
Selecting a Microcontroller (MCU)

Sensor node selection is largely a factor of the function and purpose designers are trying to fulfill. For a simple end sensor node that senses and transmits data a few times a day, an 8-bit MCU might be the correct answer. However, for advanced end nodes or gateway devices that build in intelligence or need to run an RF protocol stack or other sophisticated algorithms, a 32-bit MCU is a more appropriate choice. Some 32-bit MCUs such as those based on the ARM® Cortex®M4 core also include a floating-point unit (FPU) that proves useful for implementing complex algorithms.

The higher processing capacity of 32-bit MCUs enables them to complete processing sooner in order to enter sleep mode and preserve power. Additionally, the larger flash and RAM sizes available with 32-bit MCUs allow designers to implement the entire networking stack and application code on the MCU without needing an additional processor in the system.

ARM Cortex-M4 based 32-bit MCUs include an FPU with a DSP instruction set. Matrix multiplication, used in monitoring applications that use multiple sensors to get a more accurate reading than individual sensors, is an operation where fast single-cycle MACs (multiply-accumulate) significantly

Internet of Things System Design with Integrated Wireless MCUs 1



reduce the computation time. A common algorithm to correlate the sensor readings is the Kalman Filter, which relies heavily on matrix multiplications.

Developing applications becomes simpler with the integrated FPU. 32-bit floating-point operations are executed in hardware and the algorithms can be written directly with floating point values, taking advantage of the great dynamic range and precision of floating point numbers. Code can be more easily developed by removing the need for overflow checks needed with fixed-point processors. A good example of where ARM Cortex-M4 based MCUs are useful is when a device combines GPS, accelerometer and gyro measurements to improve location accuracy.

Selecting the Appropriate Wireless and RF Connectivity Solution

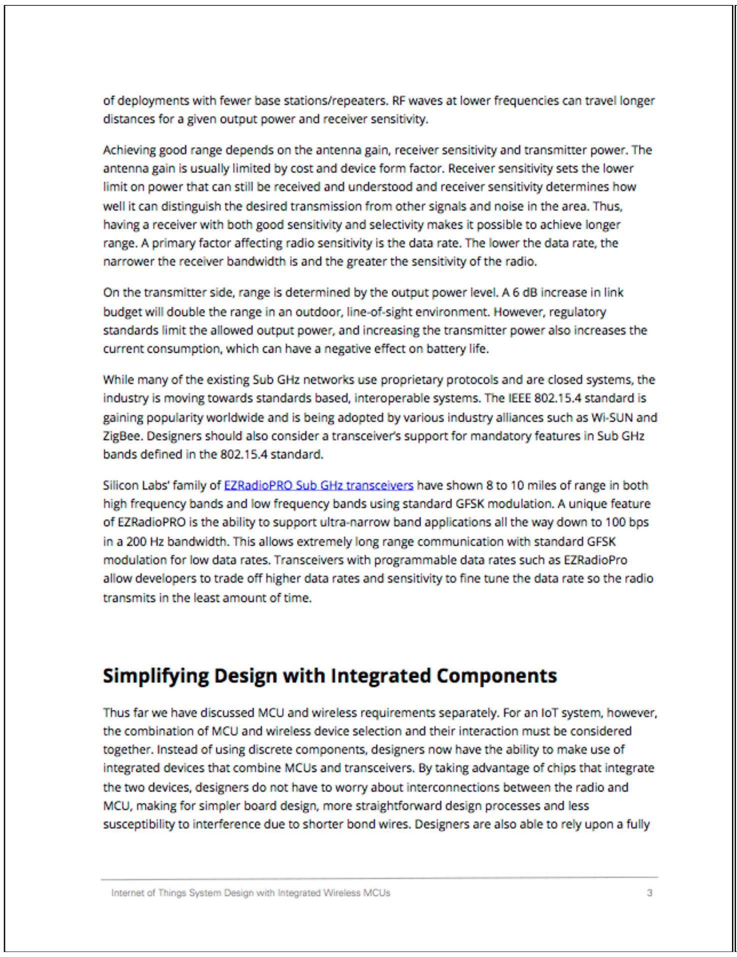
Considerations for selecting the RF protocol/technology are link budget, energy consumption and cost. Wireless developers must determine whether a Sub GHz or 2.4 GHz transceiver will best serve their application needs. Given the high profile of 2.4 GHz wireless standards such as Bluetooth® and Wi-Fi™, many manufacturers assume that 2.4 GHz is the de facto transceiver frequency of choice. Wi-Fi provides high data rate connectivity and is the most widely used protocol for bandwidth-intensive applications such as wireless cameras, while Bluetooth provides easy point-to-point connectivity with smart phones, which most people use to control their connected home applications.

Transceivers based on 2.4 GHz offer high data rates (greater than 1 Mbps) and a small antenna (less than one-third the size of a 900 MHz antenna) and are suited for short-range consumer electronics devices. However, a 2.4 GHz radio has limited range, poor wall penetration and higher power consumption. The high data rates require a wider receiver channel bandwidth, which further limits the sensitivity and range. In addition, the 2.4 GHz spectrum is crowded and subject to significant interference from Wi-Fi devices, Bluetooth nodes and microwave ovens.

Many IoT sensor nodes have relatively low data rate requirements. For these applications, sub-GHz radios offer substantially higher range than 2.4 GHz radios. Sub GHz radios also have much lower power consumption and can provide connectivity for devices that must operate for years on a single battery. These factors, combined with low system cost, make Sub GHz transceivers ideal for low data rate applications that need maximum range and multi-year battery life, such as smoke detectors, door/window sensors and outdoor systems such as weather stations and asset tracking.

One of the main advantages of using Sub GHz wireless is its long-range capability and ability to effectively pass through walls and other barriers. Narrowband transmissions can transmit data to distant hubs, often several miles away, without hopping from node to node, thus reducing the cost

Internet of Things System Design with Integrated Wireless MCUs 2



of deployments with fewer base stations/repeaters. RF waves at lower frequencies can travel longer distances for a given output power and receiver sensitivity.

Achieving good range depends on the antenna gain, receiver sensitivity and transmitter power. The antenna gain is usually limited by cost and device form factor. Receiver sensitivity sets the lower limit on power that can still be received and understood and receiver sensitivity determines how well it can distinguish the desired transmission from other signals and noise in the area. Thus, having a receiver with both good sensitivity and selectivity makes it possible to achieve longer range. A primary factor affecting radio sensitivity is the data rate. The lower the data rate, the narrower the receiver bandwidth is and the greater the sensitivity of the radio.

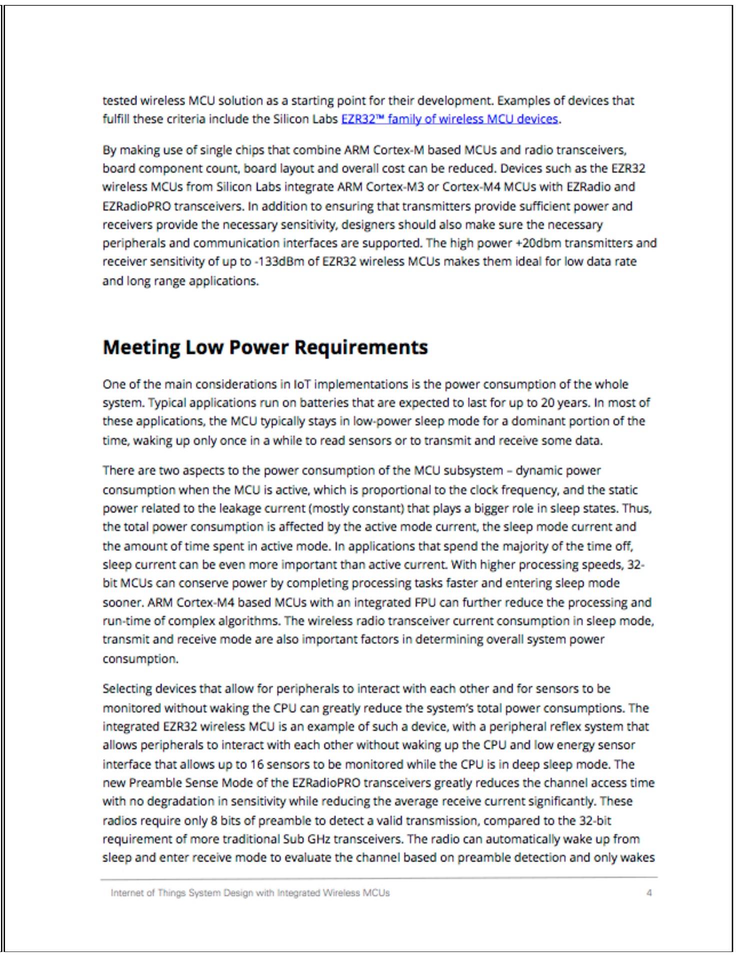
On the transmitter side, range is determined by the output power level. A 6 dB increase in link budget will double the range in an outdoor, line-of-sight environment. However, regulatory standards limit the allowed output power, and increasing the transmitter power also increases the current consumption, which can have a negative effect on battery life.

While many of the existing Sub GHz networks use proprietary protocols and are closed systems, the industry is moving towards standards based, interoperable systems. The IEEE 802.15.4 standard is gaining popularity worldwide and is being adopted by various industry alliances such as Wi-SUN and ZigBee. Designers should also consider a transceiver's support for mandatory features in Sub GHz bands defined in the 802.15.4 standard.

Silicon Labs' family of EZRadioPRO Sub GHz transceivers have shown 8 to 10 miles of range in both high frequency bands and low frequency bands using standard GFSK modulation. A unique feature of EZRadioPRO is the ability to support ultra-narrow band applications all the way down to 100 bps in a 200 Hz bandwidth. This allows extremely long range communication with standard GFSK modulation for low data rates. Transceivers with programmable data rates such as EZRadioPro allow developers to trade off higher data rates and sensitivity to fine tune the data rate so the radio transmits in the least amount of time.

Simplifying Design with Integrated Components

Thus far we have discussed MCU and wireless requirements separately. For an IoT system, however, the combination of MCU and wireless device selection and their interaction must be considered together. Instead of using discrete components, designers now have the ability to make use of integrated devices that combine MCUs and transceivers. By taking advantage of chips that integrate the two devices, designers do not have to worry about interconnections between the radio and MCU, making for simpler board design, more straightforward design processes and less susceptibility to interference due to shorter bond wires. Designers are also able to rely upon a fully



tested wireless MCU solution as a starting point for their development. Examples of devices that fulfill these criteria include the Silicon Labs EZR32™ family of wireless MCU devices.

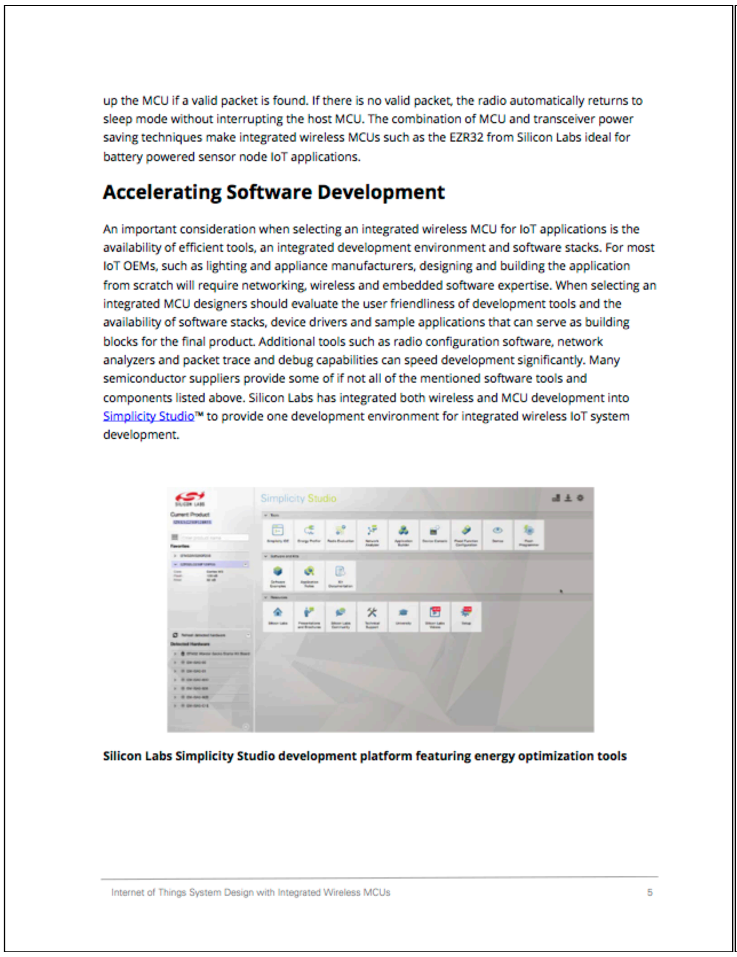
By making use of single chips that combine ARM Cortex-M based MCUs and radio transceivers, board component count, board layout and overall cost can be reduced. Devices such as the EZR32 wireless MCUs from Silicon Labs integrate ARM Cortex-M3 or Cortex-M4 MCUs with EZRadio and EZRadioPRO transceivers. In addition to ensuring that transmitters provide sufficient power and receivers provide the necessary sensitivity, designers should also make sure the necessary peripherals and communication interfaces are supported. The high power +20dbm transmitters and receiver sensitivity of up to -133dBm of EZR32 wireless MCUs makes them ideal for low data rate and long range applications.

Meeting Low Power Requirements

One of the main considerations in IoT implementations is the power consumption of the whole system. Typical applications run on batteries that are expected to last for up to 20 years. In most of these applications, the MCU typically stays in low-power sleep mode for a dominant portion of the time, waking up only once in a while to read sensors or to transmit and receive some data.

There are two aspects to the power consumption of the MCU subsystem – dynamic power consumption when the MCU is active, which is proportional to the clock frequency, and the static power related to the leakage current (mostly constant) that plays a bigger role in sleep states. Thus, the total power consumption is affected by the active mode current, the sleep mode current and the amount of time spent in active mode. In applications that spend the majority of the time off, sleep current can be even more important than active current. With higher processing speeds, 32-bit MCUs can conserve power by completing processing tasks faster and entering sleep mode sooner. ARM Cortex-M4 based MCUs with an integrated FPU can further reduce the processing and run-time of complex algorithms. The wireless radio transceiver current consumption in sleep mode, transmit and receive mode are also important factors in determining overall system power consumption.

Selecting devices that allow for peripherals to interact with each other and for sensors to be monitored without waking the CPU can greatly reduce the system's total power consumptions. The integrated EZR32 wireless MCU is an example of such a device, with a peripheral reflex system that allows peripherals to interact with each other without waking up the CPU and low energy sensor interface that allows up to 16 sensors to be monitored while the CPU is in deep sleep mode. The new Preamble Sense Mode of the EZRadioPRO transceivers greatly reduces the channel access time with no degradation in sensitivity while reducing the average receive current significantly. These radios require only 8 bits of preamble to detect a valid transmission, compared to the 32-bit requirement of more traditional Sub GHz transceivers. The radio can automatically wake up from sleep and enter receive mode to evaluate the channel based on preamble detection and only wakes

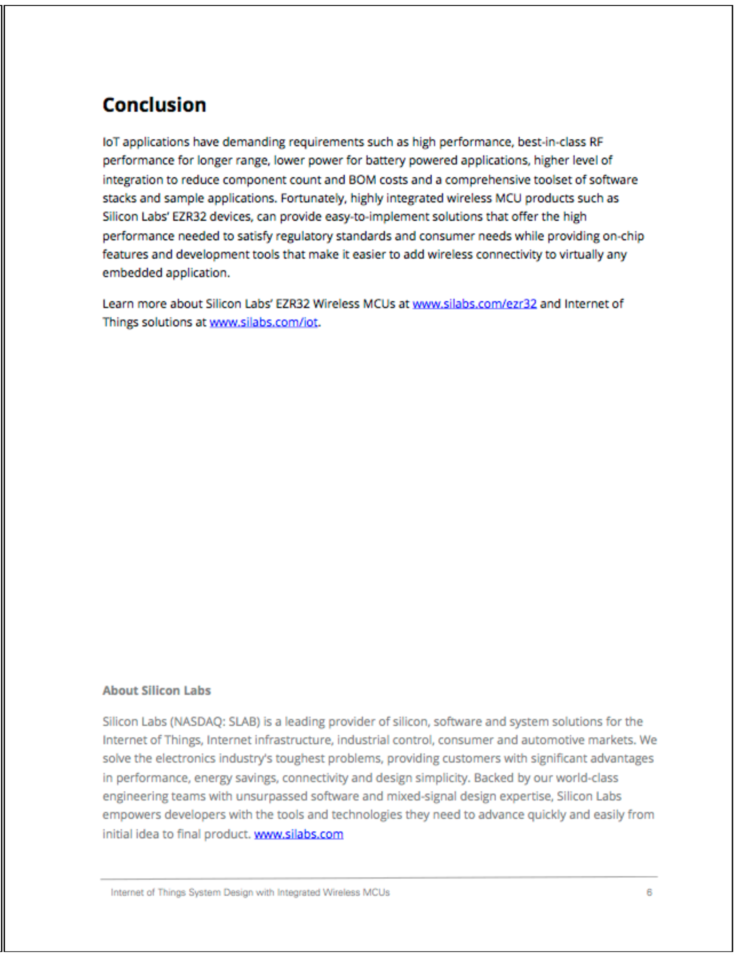


up the MCU if a valid packet is found. If there is no valid packet, the radio automatically returns to sleep mode without interrupting the host MCU. The combination of MCU and transceiver power saving techniques make integrated wireless MCUs such as the EZR32 from Silicon Labs ideal for battery powered sensor node IoT applications.

Accelerating Software Development

An important consideration when selecting an integrated wireless MCU for IoT applications is the availability of efficient tools, an integrated development environment and software stacks. For most IoT OEMs, such as lighting and appliance manufacturers, designing and building the application from scratch will require networking, wireless and embedded software expertise. When selecting an integrated MCU designers should evaluate the user friendliness of development tools and the availability of software stacks, device drivers and sample applications that can serve as building blocks for the final product. Additional tools such as radio configuration software, network analyzers and packet trace and debug capabilities can speed development significantly. Many semiconductor suppliers provide some of if not all of the mentioned software tools and components listed above. Silicon Labs has integrated both wireless and MCU development into Simplicity Studio™ to provide one development environment for integrated wireless IoT system development.

Silicon Labs Simplicity Studio development platform featuring energy optimization tools



Conclusion

IoT applications have demanding requirements such as high performance, best-in-class RF performance for longer range, lower power for battery powered applications, higher level of integration to reduce component count and BOM costs and a comprehensive toolset of software stacks and sample applications. Fortunately, highly integrated wireless MCU products such as Silicon Labs' EZR32 devices, can provide easy-to-implement solutions that offer the high performance needed to satisfy regulatory standards and consumer needs while providing on-chip features and development tools that make it easier to add wireless connectivity to virtually any embedded application.

Learn more about Silicon Labs' EZR32 Wireless MCUs at www.silabs.com/ezr32 and Internet of Things solutions at www.silabs.com/iot.

About Silicon Labs

Silicon Labs (NASDAQ: SLAB) is a leading provider of silicon, software and system solutions for the Internet of Things, Internet infrastructure, industrial control, consumer and automotive markets. We solve the electronics industry's toughest problems, providing customers with significant advantages in performance, energy savings, connectivity and design simplicity. Backed by our world-class engineering teams with unsurpassed software and mixed-signal design expertise, Silicon Labs empowers developers with the tools and technologies they need to advance quickly and easily from initial idea to final product. www.silabs.com

Annexe 6 | Intel — The Microprocessor Revolution

Identification

- Adresse du site : *http://www.i-programmer.info/history/machines/734-intel.html*
- Nom du site : I Programmer
- Date de publication : 5 août 2010
- Date de consultation : 29 mars 2017

Évaluation du contenu

Critère d'évaluation	Faible	Moyen	Bien	Très bien
Degré de fiabilité de l'informatique			X	
Utilité, accessibilité, pertinence de ces informations			X	
Utilité des illustrations			X	
Qualité, nombre…				X

Mots clés

- Intel
- Microprocesseur
- Révolution
- Composants
- Bits
- Compact
- Prix
- Instructions
- Puissance

Compte rendu

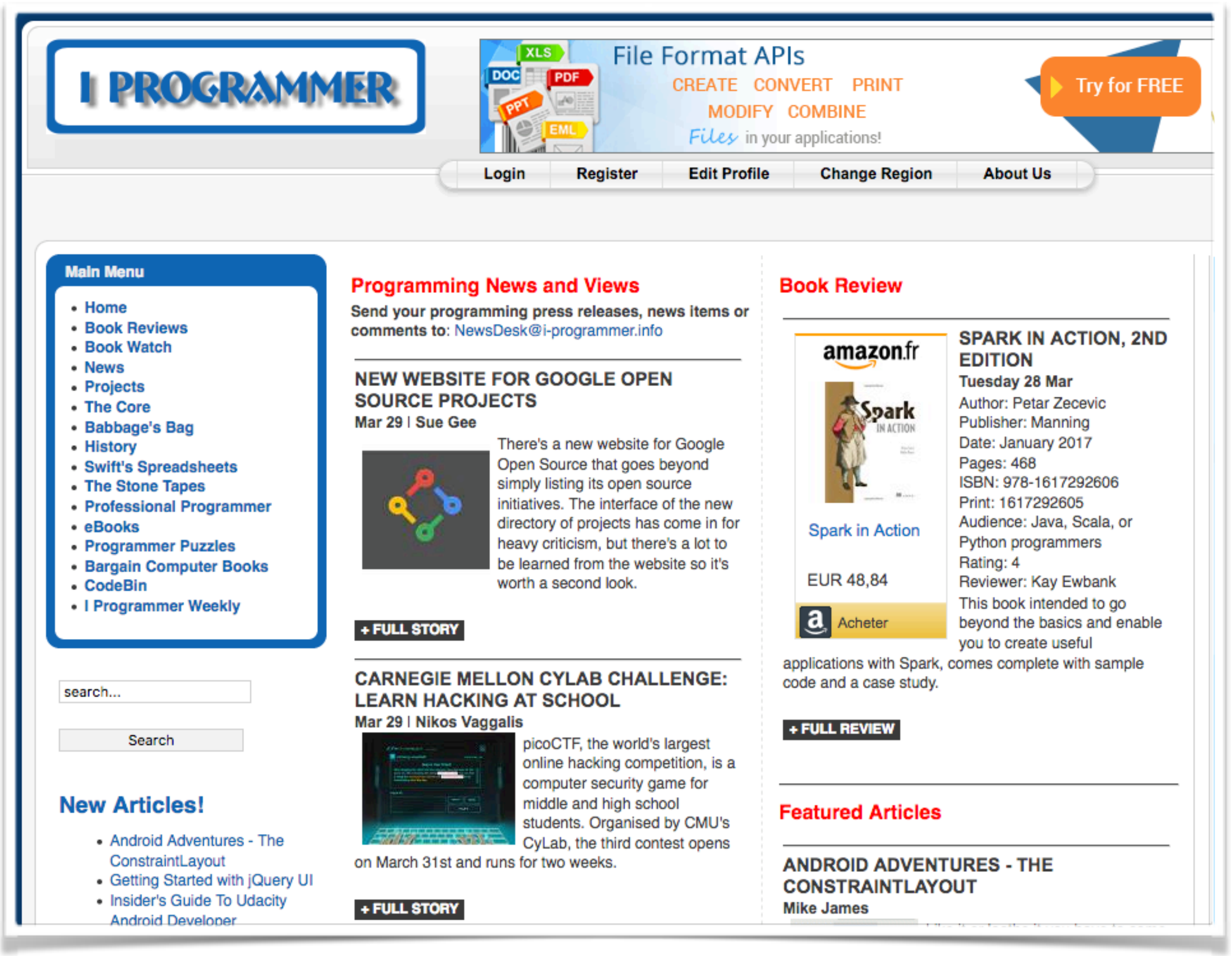
Cet article est un **historique** du microprocesseur dans les grandes lignes et notamment de l'importante participation d'**Intel** dans l'élaboration et la conception de ceux-ci. En 1965, Gordon Moore exposa sa théorie : chaque année, las circuits informatiques contiendraient **deux fois plus** de composants que l'année précédente. Cette théorie se révéla toujours vraie jusqu'à il y a quelques années.

Hoff, Shima et Faggin sont les trois hommes à l'**origine** de l'Intel 4004, le **premier** microprocesseur au monde. Shima et Faggin ont ensuite quitté Intel pour fonder **Zilog**, qui sorti en 1976 le Z80, un microprocesseur très populaire, utilisé dans des ordinateurs tels que les TRS-80, ZX Spectrum ou encore les Amstrad CPC.

Plus tard chez Intel, en 1972, le premier microprocesseur **8 bits** a été conçu : l'Intel 8008. Il est dit que si l'on plaçait 20 autres circuits intégrés avec un Intel 8008, on avait une machine que l'on pouvait appeler « **ordinateur** ».

Avis

Cet historique montre bien l'impact qu'a eu le microprocesseur dans le monde de l'informatique moderne et comment ils ont été conçus. Les processeurs 8 bits apparu il y a plus de 40 ans sont encore majoritairement utilisés aujourd'hui et ne changent plus beaucoup. Cet article pourrait bénéficier de plus de détails, notamment sur les évolutions techniques qu'il y a eu entre les différentes catégories de microprocesseur apparues au fil des années. Je lui attribue la note de 7/10.



New Articles!

- Android Adventures - The ConstraintLayout
- Getting Started with jQuery UI
- Insider's Guide To Udacity Android Developer

Programming News and Views

Send your programming press releases, news items or comments to: NewsDesk@i-programmer.info

NEW WEBSITE FOR GOOGLE OPEN SOURCE PROJECTS

Mar 29 | Sue Gee

There's a new website for Google Open Source that goes beyond simply listing its open source initiatives. The interface of the new directory of projects has come in for heavy criticism, but there's a lot to be learned from the website so it's worth a second look.

CARNEGIE MELLON CYLAB CHALLENGE: LEARN HACKING AT SCHOOL

Mar 29 | Nikos Vaggalis

picoCTF, the world's largest online hacking competition, is a computer security game for middle and high school students. Organised by CMU's CyLab, the third contest opens on March 31st and runs for two weeks.

Book Review

Spark in Action

EUR 48,84

SPARK IN ACTION, 2ND EDITION

Tuesday 28 Mar

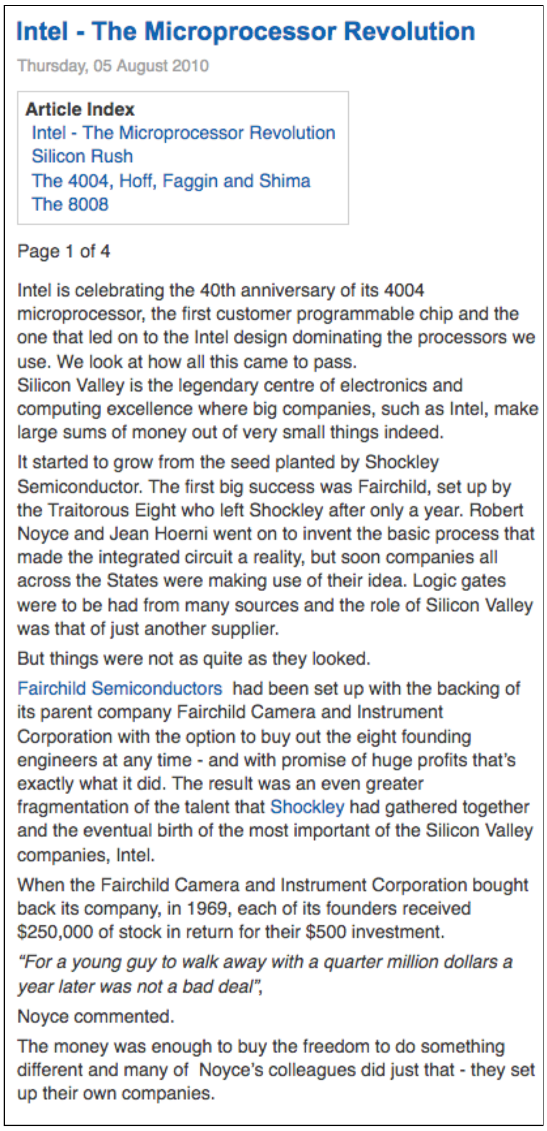
Author: Petar Zecevic
Publisher: Manning
Date: January 2017
Pages: 468
ISBN: 978-1617292606
Print: 1617292605
Audience: Java, Scala, or Python programmers
Rating: 4
Reviewer: Kay Ewbank

This book intended to go beyond the basics and enable you to create useful applications with Spark, comes complete with sample code and a case study.

Featured Articles

ANDROID ADVENTURES - THE CONSTRAINTLAYOUT

Mike James



Intel - The Microprocessor Revolution

Thursday, 05 August 2010

Article Index

Page 1 of 4

Intel is celebrating the 40th anniversary of its 4004 microprocessor, the first customer programmable chip and the one that led on to the Intel design dominating the processors we use. We look at how all this came to pass.
Silicon Valley is the legendary centre of electronics and computing excellence where big companies, such as Intel, make large sums of money out of very small things indeed.

It started to grow from the seed planted by Shockley Semiconductor. The first big success was Fairchild, set up by the Traitorous Eight who left Shockley after only a year. Robert Noyce and Jean Hoerni went on to invent the basic process that made the integrated circuit a reality, but soon companies all across the States were making use of their idea. Logic gates were to be had from many sources and the role of Silicon Valley was that of just another supplier.

But things were not as quite as they looked.

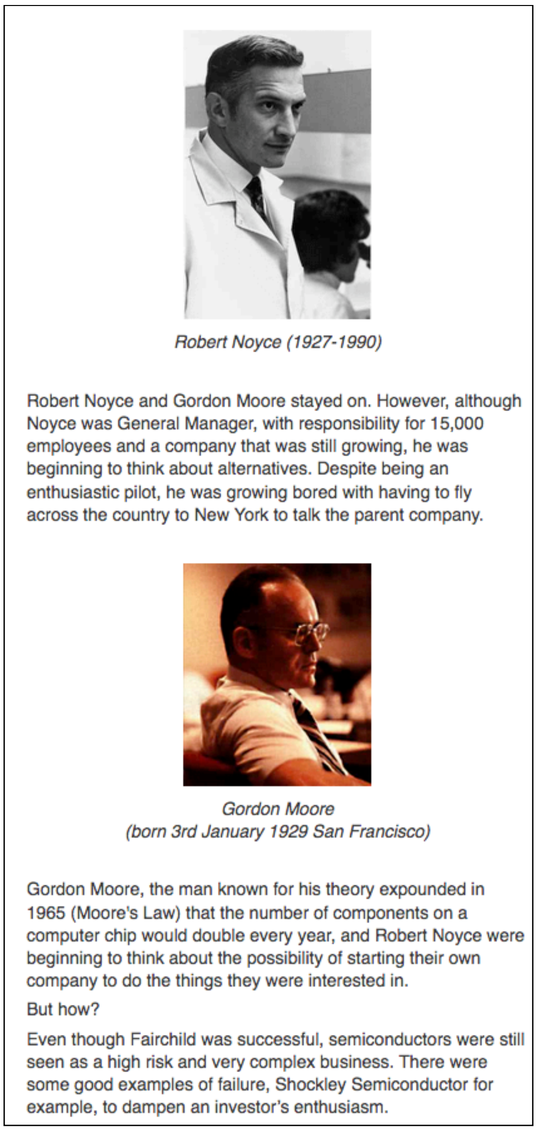
Fairchild Semiconductors had been set up with the backing of its parent company Fairchild Camera and Instrument Corporation with the option to buy out the eight founding engineers at any time - and with promise of huge profits that's exactly what it did. The result was an even greater fragmentation of the talent that Shockley had gathered together and the eventual birth of the most important of the Silicon Valley companies, Intel.

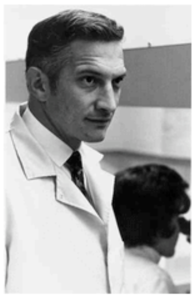
When the Fairchild Camera and Instrument Corporation bought back its company, in 1969, each of its founders received $250,000 of stock in return for their $500 investment.

"For a young guy to walk away with a quarter million dollars a year later was not a bad deal",

Noyce commented.

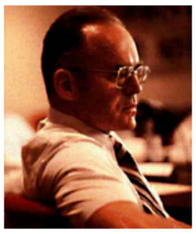
The money was enough to buy the freedom to do something different and many of Noyce's colleagues did just that - they set up their own companies.





Robert Noyce (1927-1990)

Robert Noyce and Gordon Moore stayed on. However, although Noyce was General Manager, with responsibility for 15,000 employees and a company that was still growing, he was beginning to think about alternatives. Despite being an enthusiastic pilot, he was growing bored with having to fly across the country to New York to talk the parent company.

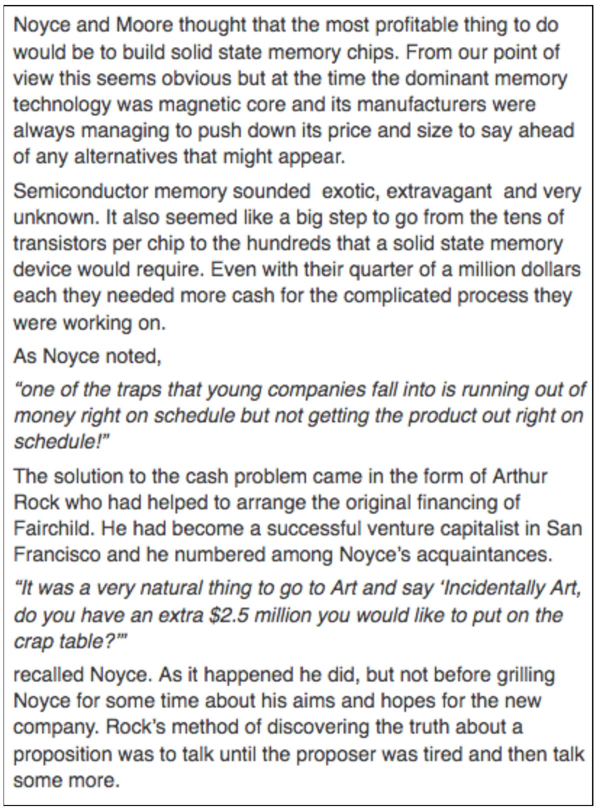


Gordon Moore
(born 3rd January 1929 San Francisco)

Gordon Moore, the man known for his theory expounded in 1965 (Moore's Law) that the number of components on a computer chip would double every year, and Robert Noyce were beginning to think about the possibility of starting their own company to do the things they were interested in.

But how?

Even though Fairchild was successful, semiconductors were still seen as a high risk and very complex business. There were some good examples of failure, Shockley Semiconductor for example, to dampen an investor's enthusiasm.



Noyce and Moore thought that the most profitable thing to do would be to build solid state memory chips. From our point of view this seems obvious but at the time the dominant memory technology was magnetic core and its manufacturers were always managing to push down its price and size to say ahead of any alternatives that might appear.

Semiconductor memory sounded exotic, extravagant and very unknown. It also seemed like a big step to go from the tens of transistors per chip to the hundreds that a solid state memory device would require. Even with their quarter of a million dollars each they needed more cash for the complicated process they were working on.

As Noyce noted,

"one of the traps that young companies fall into is running out of money right on schedule but not getting the product out right on schedule!"

The solution to the cash problem came in the form of Arthur Rock who had helped to arrange the original financing of Fairchild. He had become a successful venture capitalist in San Francisco and he numbered among Noyce's acquaintances.

"It was a very natural thing to go to Art and say 'Incidentally Art, do you have an extra $2.5 million you would like to put on the crap table?'"

recalled Noyce. As it happened he did, but not before grilling Noyce for some time about his aims and hopes for the new company. Rock's method of discovering the truth about a proposition was to talk until the proposer was tired and then talk some more.