

SEE – Introduction générale

Cours - Master 2 LSE

Jalil Boukhobza

Université de Bretagne Occidentale – Lab-STICC

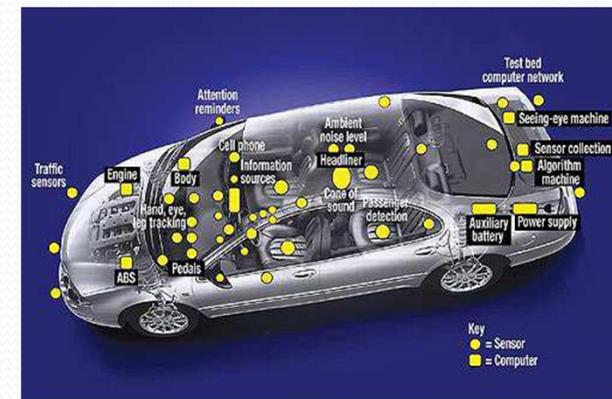
Qu'est ce qu'un système embarqué

- **Ébauche d'une définition:** c'est un système électronique et informatique autonome qui est dédié à une tâche particulière et contenue dans un système englobant. Il n'est « généralement » pas programmable.
 - Pas d'E/S standards
 - Matériel et application intimement liés
 - Logiciel enfoui ... noyé dans le matériel ... pas facilement discernable comme dans un PC.



- **Il sont partout !!!**

- Radio/réveil
- Machine à café
- Télévision / télécommande
- Moyen de transport (voiture : à foison !)
- Téléphone portable



Caractéristiques principales d'un système embarqué

- Système principalement **numérique**
- Met généralement en œuvre un **processeur**
- Exécute une application logicielle dédiée précise (non pas une application scientifique ou grand public → tend à être moins vrai)
- Il n'a pas réellement de clavier standard et l'affichage est limité
- Ce ne sont pas des PC, mais des architectures similaires (x86) basse consommation (l'une des contraintes ...).

L'embarqué en quelques chiffres

- **En 1999**, il a été vendu (**dans le domaine de l'embarqué**):
 - 1,3 **milliards** de processeurs 4 bits
 - 1,4 **milliards** de processeurs 8 bits
 - 375 millions de processeurs 16 bits
 - 127 millions de processeurs 32 bits
 - 3,2 millions de processeurs 64 bits.
- Il a été vendu **108 millions** de processeurs pour le marché du PC
- **En 2004**:
 - 14 milliards de processeurs pour l'embarqué (microprocesseur, SoC, DSP, etc.)
 - 260 millions de processeurs PC.

- Moins de 2% (5%) des processeurs vendus sont pour les PC, 98% (95%) pour l'embarqué
- Prix moyen d'un processeur 6\$ (2004) alors qu'un processeur PC coûte 300\$.

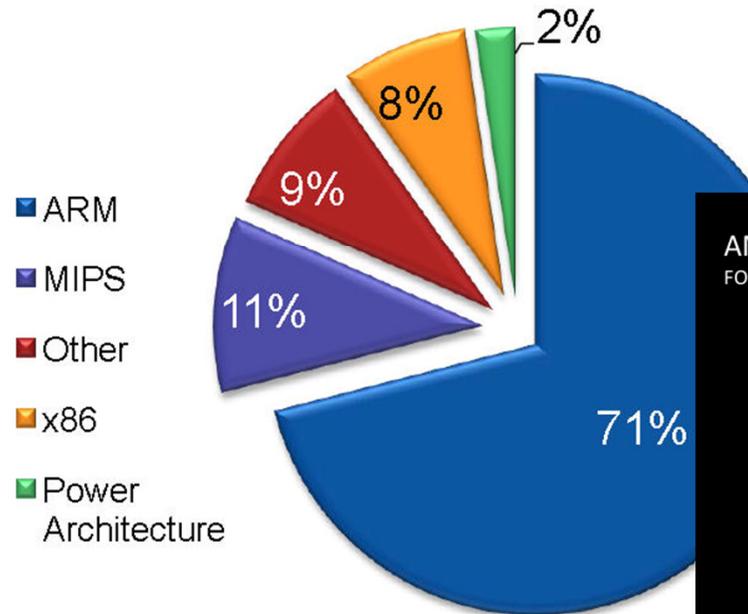


- **Le marché du processeur PC est de très faible volume ... mais très lucratif**



Source: <http://chipdesignmag.com/sld/schirrmeister/2012/04/24/processor-architecture-enablement-betting-on-the-right-processor/>

2011 CPU Unit Shipments by Architecture



2011 CPU Revenue by Architecture



AMD EMBEDDED SOLUTIONS
FOCUS AREAS

AMD

2013 TAM \$11.3B

Architecture	Percentage
x86	44%
ARM	36%
MIPS	7%
Power	6%
Other	7%

2016 TAM \$15.5B

Architecture	Percentage
x86	45%
ARM	37%
MIPS	6%
Power	6%
Other	6%

X86 & ARM ACCOUNT FOR 82% OF \$15.5B TAM BY 2016

Champs d'application: 4 grandes classes

1. Calcul généraliste

- Similaire aux applications bureau mais embarqué (assistant personnel, téléphone portable, tablettes, etc.)
- Consoles de jeux vidéo, set-top box

2. Contrôle de systèmes

- Moteur, voiture, avion, processus chimique, nucléaire, navigation, etc.

3. Traitement du signal

- Compression vidéo, radar, flux de données, etc.

4. Réseaux et communications

- Transmission de données, commutation, routage, téléphone, Internet, etc.

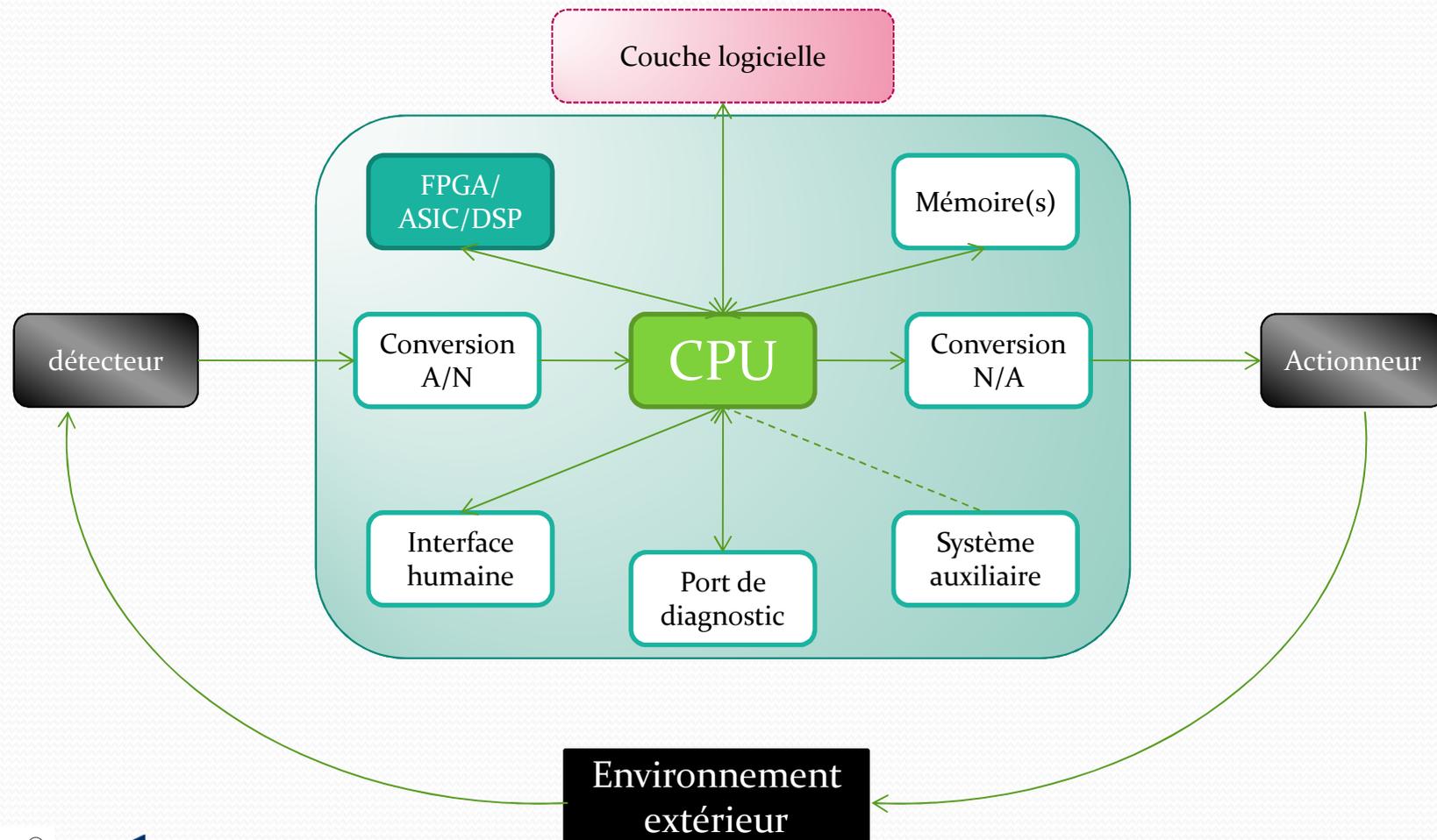
Exemples de systèmes embarqués

- **Grand public :**
 - Appareils photographiques et caméras, lecteurs DVD, chauffage et climatisation, éclairage, électroménager, domotique, sécurité (incendie, intrusion, surveillance, piscine), ascenseurs, HiFi, audio et vidéo, consoles de jeux, décodeurs, etc.
- **Transports :**
 - Automobile, aéronautique, spatial, marine, assistance à la conduite ou au pilotage, maintenance, signalisation, contrôle du trafic aérien, maritime (aujourd'hui aide, demain automatique, objectif trafic autoroutier), distributeur de billets, radar, etc.
- **Défense :**
 - Contrôle de trajectoire, lanceur, etc.

Exemples de systèmes embarqués (2)

- **Secteur manufacturier et industrie :**
 - Chaînes de production, automates, production et distribution d'électricité, réacteurs chimiques, réacteurs nucléaires, raffineries, dispositifs de sécurité, aide à la maintenance, etc.
- **Information et communication :**
 - Imprimante, périphérique, téléphone, répondeur, fax, routeurs, téléphonie mobile, satellites, GPS, etc.
- **Santé :**
 - Imagerie médicale, diagnostique, soins, implants, handicapés, etc.
- **Autres :**
 - Carte à puce, distributeurs, etc.

Systeme embarqué « typique »



Systeme embarqué « typique » (2)

- **Capteurs** (interrupteurs, etc.) couplés à des convertisseurs analogique/numérique.
- **Actionneurs** (LED, etc.) couplés à des convertisseurs numérique/analogique.
- **Calculateur** (processeur embarqué et ses E/S).
- Possibilité d'avoir un/des **FPGAs et/ou ASICs et/ou DSP** pour jouer le rôle de coprocesseurs (accélération matérielle)

- Les systèmes embarqués doivent prendre en compte:
 - Variation des températures
 - Vibrations et chocs
 - Variations des alimentations
 - Interférences RF
 - Corrosion
 - Eau, feu, radiation
 - ...

Quelques propriétés des systèmes embarqués

- **Ciblé**: domaine d'action limité aux fonctions pour lesquelles il a été créé.
- **Simple**: gage de robustesse
- **Fiable**: fonctionnement complètement autonome.
- **Sécurisé**
- **Maintenable** dans le temps: certains produits sont censés durer jusqu'à 20 ans et plus (surtout dans le domaine militaire)
- **Interface spécifique**: approche matérielle à cause de contraintes d'optimisation
- **Optimisé**: généralement logicielles de petite taille car plus c'est grand, plus il y a de chance d'avoir des bugs. Ce sont des logiciels produits à grande échelle, le moindre centime compte.
- **Tolérant aux fautes**

Métriques et caractéristiques des systèmes embarqués

- **Puissance de calcul**: sélection du processeur d'après la charge de travail à effectuer et d'après la largeur des registres.
- **Débit**: le système peut avoir besoin de prendre en charge plusieurs données en même temps.
- **Temps de réponse**: le système doit réagir vite
- **L'empreinte mémoire**: l'estimation la plus précise possible de l'utilisation mémoire
- **Consommation d'énergie**: la conception hw ET sw doit prendre en compte la conso
- **Nombre d'unités produites** dicte le compromis entre prix de production et prix de développement
- **Durée de vie** est un paramètre important lors de la décision des composants à inclure
- **Installation du programme** requiert des outils spécifiques
- **Testabilité et débogage** plus dur sans E/S de type clavier, etc.
- **Fiabilité** est un paramètre critique ex: système ABS

Autres métriques

- Coût à l'unité
- Coût NRE (Non Recurring Engineering)
- **Flexibilité**: changer la fonctionnalité sans coût NRE trop lourd
- Time-to-prototype
- Time-to-market
-

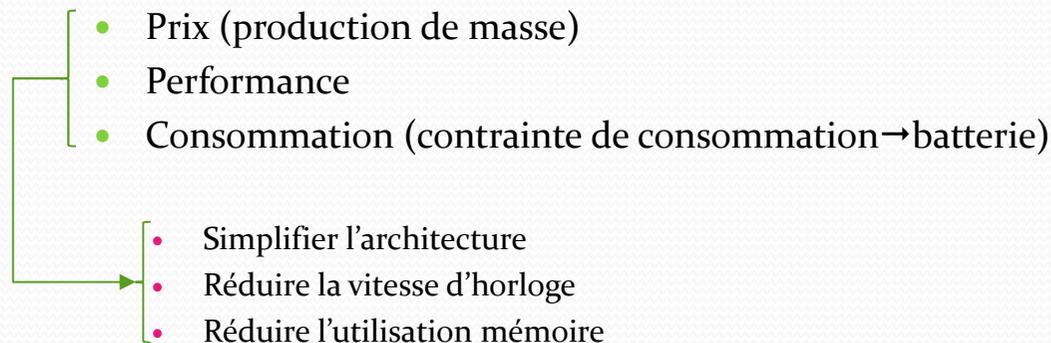
Logiciel/matériel embarqué

- **Logiciel embarqué:** programme/application utilisé dans un équipement et complètement intégré dans ce dernier.
- **Système embarqué:** Matériel(s) + logiciel(s) (+ OS)

2 types de systèmes embarqués (UNE classification):

1. **Systèmes embarqués destinés à l'utilisateur (high-end):** généralement une version dégradée d'un OS existant (ex: Linux). Ex: routeurs, smartphone, etc.
2. **Systèmes embarqués profondément enfouis:** peu de fonctions, très petite **empreinte mémoire**, généralement construit *from scratch*. Appareil photo numérique, téléphones portables, etc.

- Différences avec les « machines normales »:

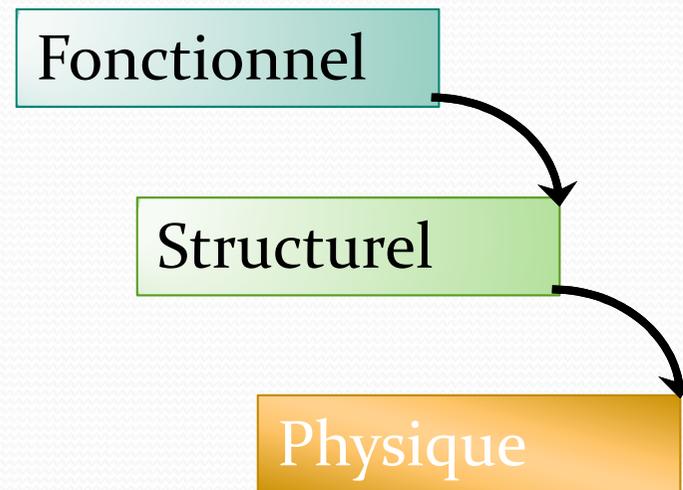


Modèles de développement

- Décrit l'ordre des étapes de conception
 - Description **fonctionnelle/comportementale**
 - Traduction fonctionnelle → **structurelle**
 - Mapping de la structure vers une **implémentation physique**
- 2 modèles (plus en réalité)
 - Modèle en cascade
 - Modèle en spirale

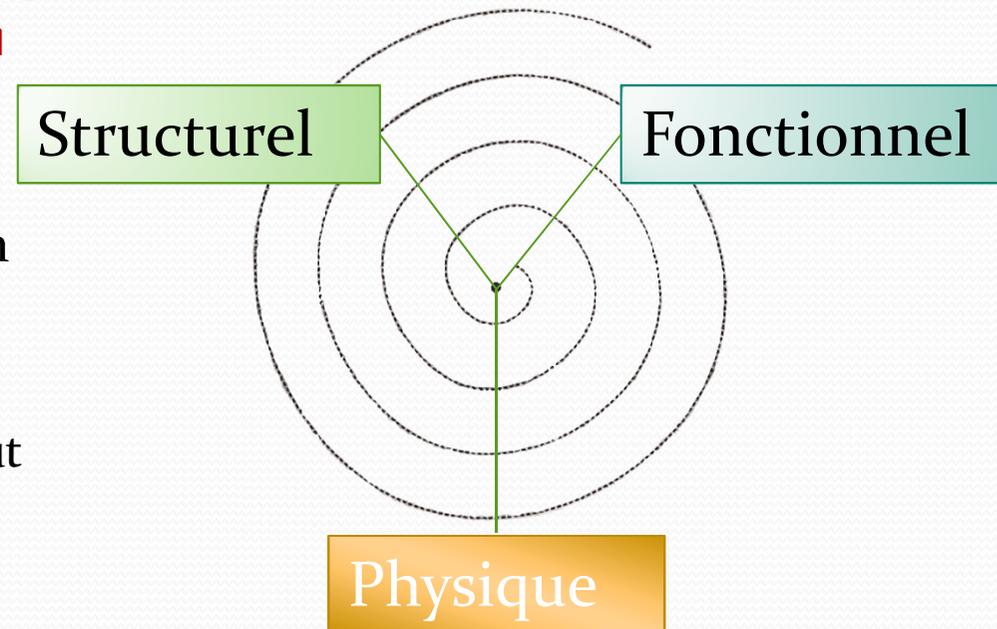
Modèle en cascade

- Pas très réaliste
 - Pb trouvés à la dernière étape → et qui nécessite une solution au niveau supérieur
 - Prototype généralement requis pour connaître l'ensemble du comportement demandé (ajout de fonctionnalité lors de la démo)
 - Les spécifications systèmes peuvent évoluer
- Retour en arrière (une quelconque mise à jour)
 - Perte de revenus
 - Mauvais positionnement dans le marché (perte de temps)



Modèle en spirale

- 1^{ère} itération sur les 3 étapes (incomplète) mais
 - Fini par un prototype (utilisé pour le test) → **time-to-prototype** ⚡
- Doit utiliser des moyens/outils permettant d'obtenir une structure et une implémentation rapidement.
 - FPGA par ex.
 - Utilisation de **plus d'outils** (coût plus important)
- Plus couteux en temps que le modèle en cascade (si toutefois sa 1^{ère} itération est bonne)



Modèles et langages

- Comment « capturer » le comportement d'un sys. Embarqué ?
 - Ne pas penser langage → penser **modèle de calcul**
- **Modèles de calcul classiques:**
 - Programme séquentiel
 - Processus communicants (plusieurs prog. séquentiels)
 - Diagramme d'états (FSM) (système orienté contrôle)
 - Modèle orienté données (dataflow) (système orienté donné)
 - Modèle orienté objets (diminuer la complexité et mieux structurer)

Modèles Vs Langages



- Modèles de calcul décrivent le comportement
 - Notions conceptuels ex: recette est un prog. Séquentiel
- Langages implémentent les modèles de calcul
 - Français pour la recette, et C pour le prog. Séquentiel
- Plusieurs langages peuvent être utilisés pour un seul modèle:
 - Prog. Séq.: C, C++, Java
- Un langage pour implémenter plusieurs modèles:
 - C++: prog. séq., OO, FSM
- Certains langages sont plus adéquats que d'autre pour décrire un modèle
 - C → OO (possible, mais pas optimal)

Un Langage approprié pour un Modèle approprié

- Trouver le bon modèle de calcul pour un système donné est très important
 - Le modèle influence la façon de penser un système
 - Une machine à café → plus facilement modélisable avec un FSM plutôt qu'en OO ou en prog. Séq.
- Le langage doit « capturer » facilement le modèle
 - Avoir des facilités/constructions permettant de traduire directement les éléments du modèle vers le langage
 - Send/receive bloquant pour des tâches communicantes

L'implémentation/mise ne œuvre

- Mapping d'une fonctionnalité
→ implémentation matérielle
 - Modélisé avec un modèle de calcul donné
 - Écrite dans un langage donné
- Les choix d'implémentation doivent être (le plus) indépendant (possible) des choix de langage → **outils**
- Choix d'implémentation hw
 - Consommation énergétique, contraintes physiques, taille, performance, coût, etc.

FSM OO dataflow

Est-ce que ça permet de **décrire** le comportement du système ?

C VHDL Java

Est-ce que ça permet de **traduire** le modèle de calcul

Imp. 1 Imp. 2 Imp. 3

Est-ce que ça **réponds aux contraintes** « physiques » : taille, conso, perf., etc.

3 technologies clés pour les sys. embarqués

- Technologie (source: Larousse):
 - « *Étude des outils, des machines, des procédés et des méthodes employés dans les diverses branches de l'industrie* »
- 3 technos clés:
 - **Technologie du circuit intégré (IC)**
 - **Technologie du (ou des) processeur (s)**
 - **Technologie de conception (design)**



Intérêts
pour le
M2LSE

1. Technologie du circuit intégré (IC)

- La manière avec laquelle une implémentation numérique (portes logiques) est *mappée* sur un circuit intégré
 - IC: circuit intégré (puce, *chip*)
 - Les technologies de circuits intégrés diffèrent de par leur façon d'implémenter un « design »
 - Classification:
 - Selon la techno de transistor utilisée: bipolaire (RTL, TTL, etc.)/à effet de champs (MOS, CMOS), type de boîtier, gamme de température, caractéristiques électrique, etc.

1. Technologie du circuit intégré (IC)

- Choix d'implémentation:
 - **Full custom VLSI** ~millions
 - Grande flexibilité, complexité, performance, coût (temps et argent) → GPP et certains ASIC
 - **Standard cell** ~10000
 - Utilisation d'une librairie de cellules standards + outil de placement routage automatique, création d'une nouvelle puce
 - **Gate Arrays** ~1000
 - Librairie de cellules standards, pas de création de puce
 - **PLD** (Programmable Logic Device) ~100

Semi custom ASIC

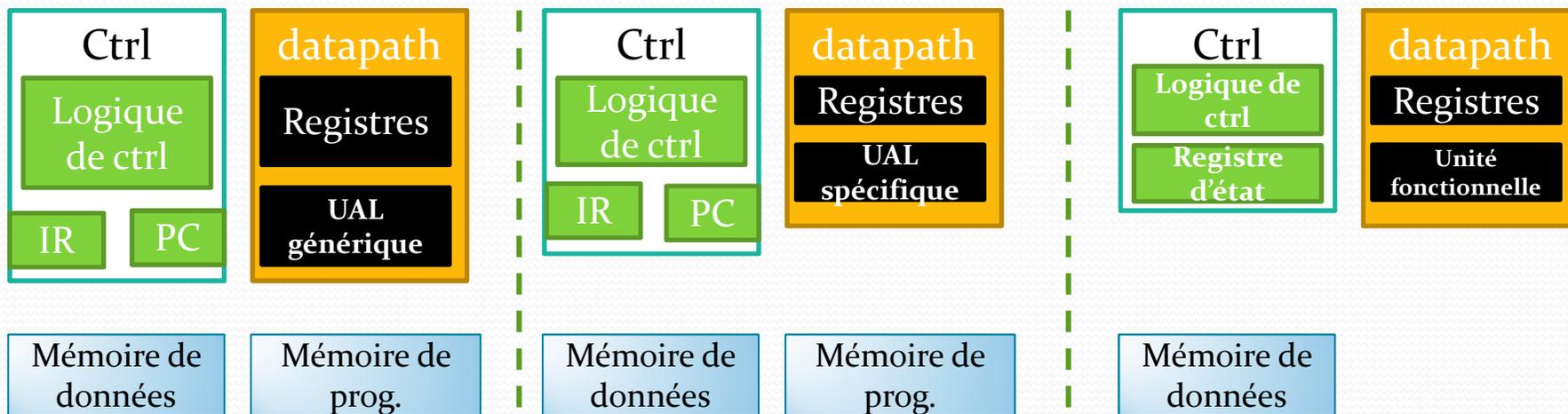
Intégration, complexité, coût, performance

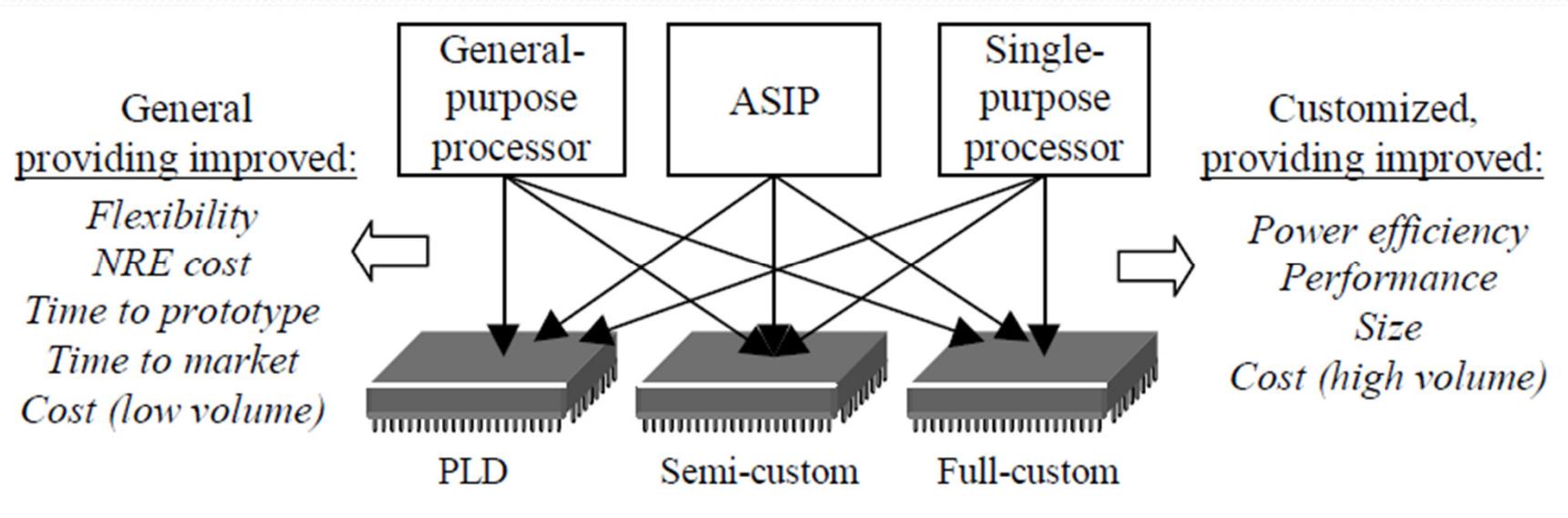
2. Technologie du processeur

- L'architecture du « calculateur » censé exécuter la fonctionnalité.
- Le processeur n'est pas forcément programmable
 - Processeur ≠ processeur généraliste (GPP)

- Flexibilité
- Coût NRE
- Time to prototype
- Time to market
- Coût (petit volume)

- Efficacité énergétique
- Performance
- Taille
- Coût (grand volume)

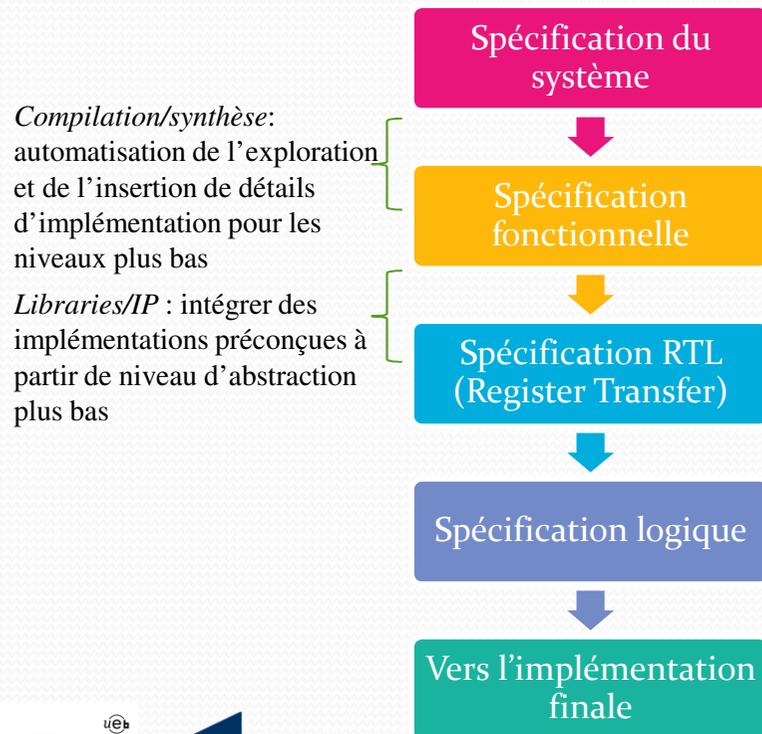




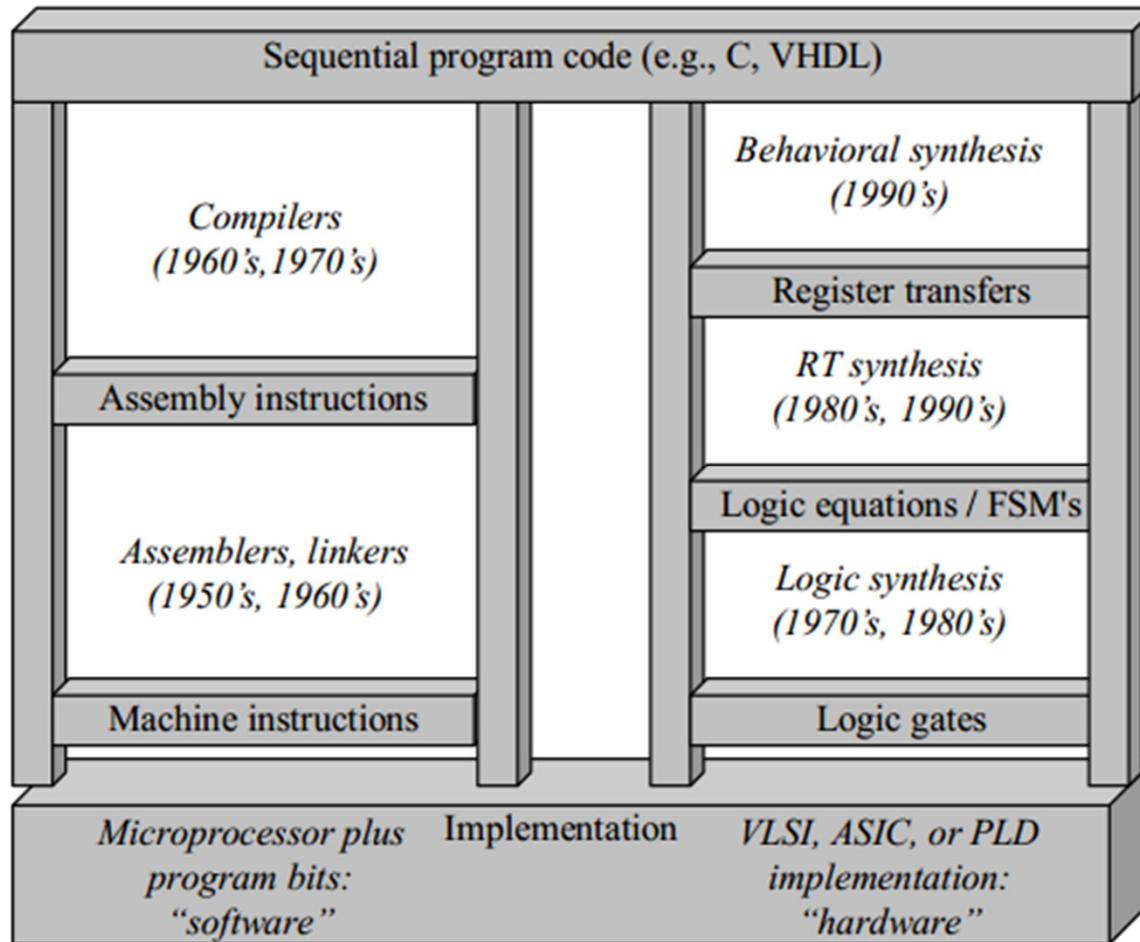
Source: Embedded System Design: A Unified Hardware/Software Approach
 Frank Vahid and Tony Givargis

3. Technologie de conception

Comment on convertie une fonctionnalité en implémentation? + comment améliorer ?



Compilation / synthèse	Librairies/IP	Test/Vérification
Synthèse système	Hw/Sw/OS	Simulation/vérif
Synthèse comportementale	Cœurs de calcul	Cosimulation Hw/Sw
Synthèse RT	Composants RT	Simulateurs HDL
Synthèse logique	Portes/cellules	Simulateurs niveau portes



Source: « Embedded System Design », Vahid/Givargis

Quelques chiffres sur le domaine de l'embarqué

Source:

<http://m.eet.com/media/1246048/2017-embedded-market-study.pdf>



ASPENCORE

2017 Embedded Markets Study

**Integrating IoT and Advanced Technology Designs, Application
Development & Processing Environments**

April 2017

Presented By: **EETimes** embedded

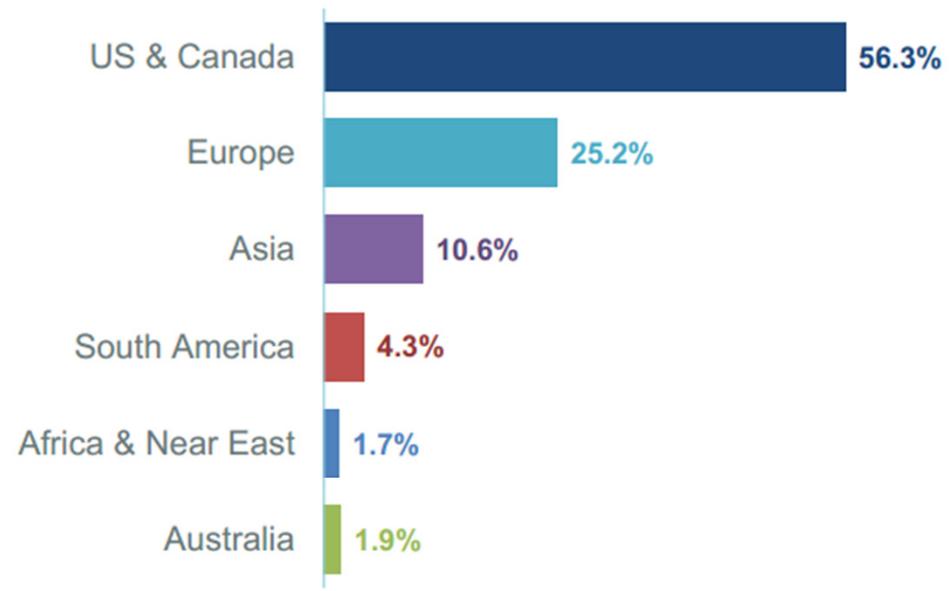
Méthodologie

- **Purpose:** To profile the findings of the 2017 results of EETimes/Embedded.com comprehensive survey of the **embedded systems markets worldwide**. Findings include types of technology used, all aspects of the embedded development process, IoT and innovative technologies emergence, tools used, work environment, applications, methods/ processes, operating systems used, reasons for using chips and technology, and brands and chips being considered by embedded developers. Many questions in this survey are trended over three to five years, but in 2016 the survey was not conducted, so there is no data from that year.
- **Methodology:** **A web-based online survey** instrument based on the 2015 annual survey was developed and implemented by independent research company Wilson Research Group on February 20, 2017 through to April 15, 2017 by email invitation.
- **Sample:** E-mail invitations were sent to subscribers to EETimes and Embedded.com and related brands with reminder invitations sent later. Each invitation included a link to the survey and an incentive to participate.
- **Returns:** **Data is based on 1,234 valid respondents** for an overall confidence of 95% +/- 2.8%. Confidence levels vary by question.



ASPENCORE

In which region of the world do you reside?



Idée reçue numéro 1

Les systèmes embarqués, c'est pour les électroniciens c'est bien connu ...!

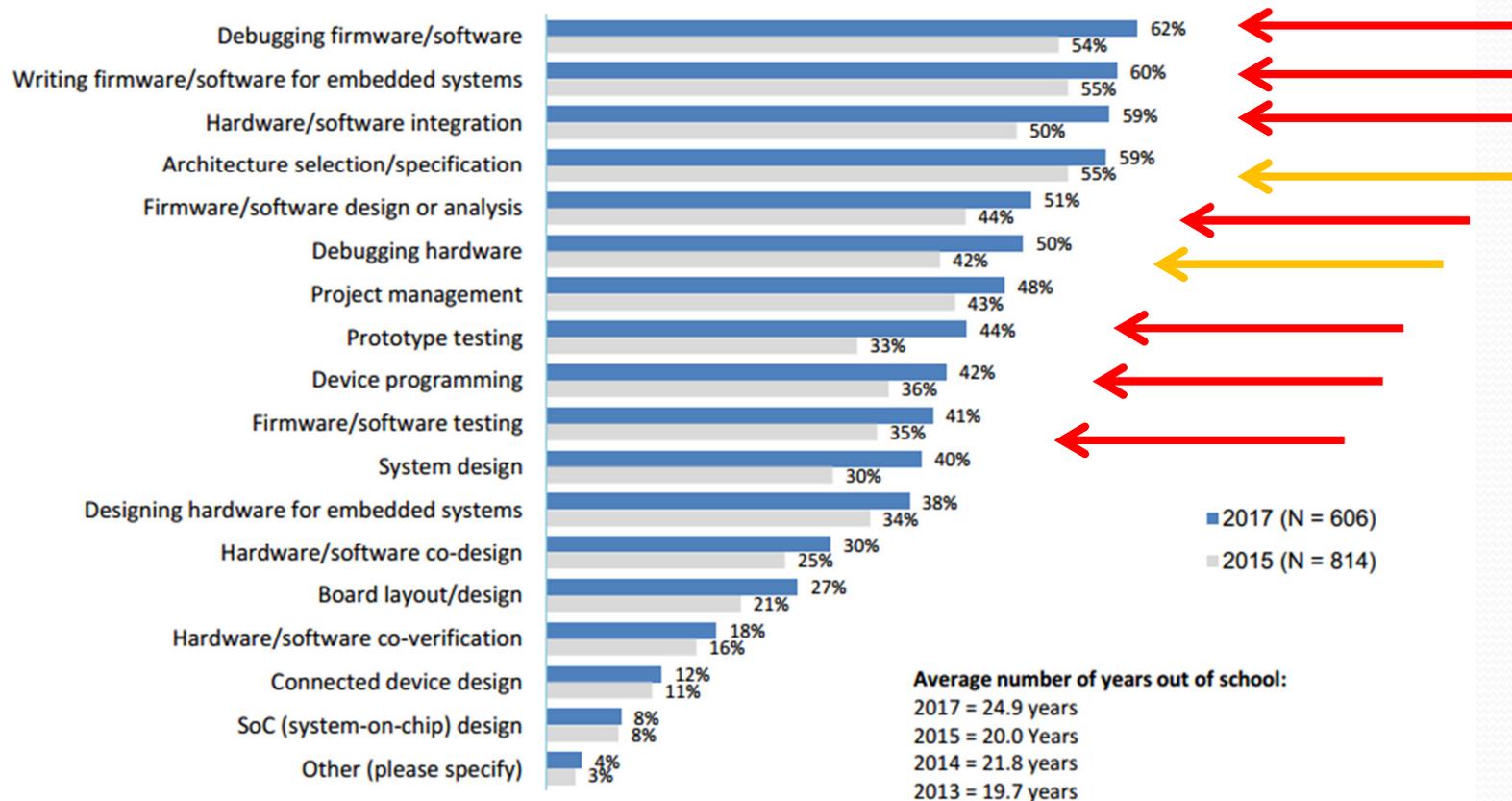
MR.GEEK





ASPENCORE

Job Functions

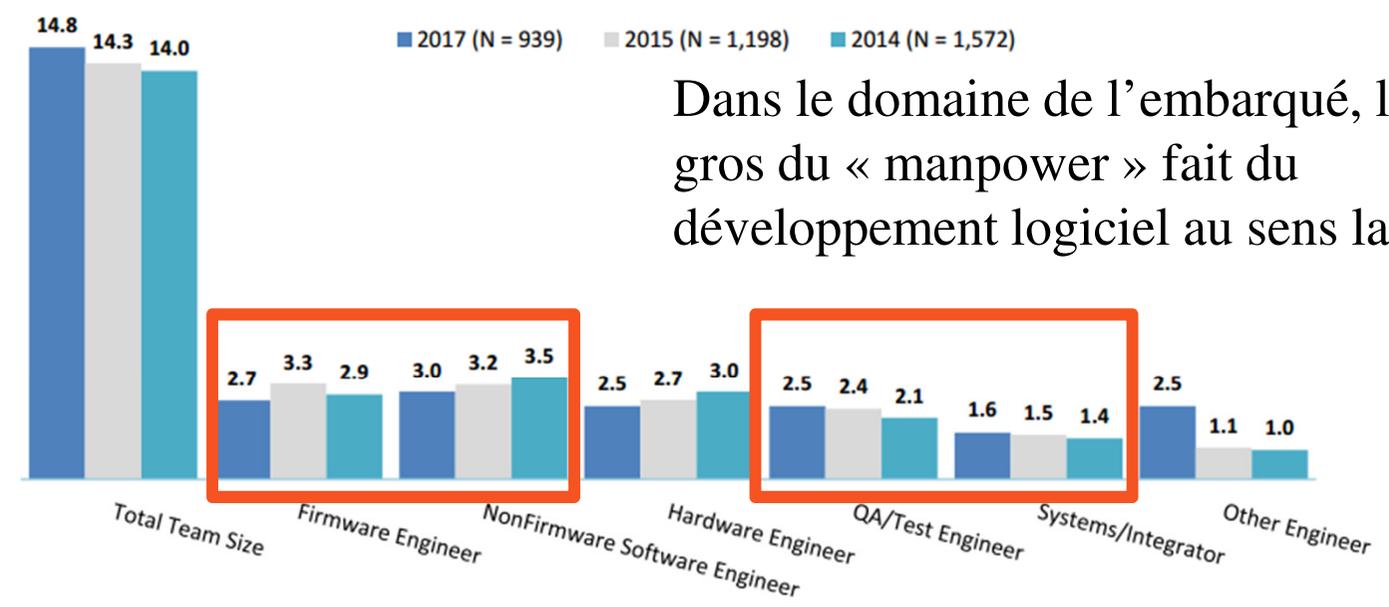




How many people are on your embedded project team?



14.8 engineers per team is slightly higher than 2015 and 2014.

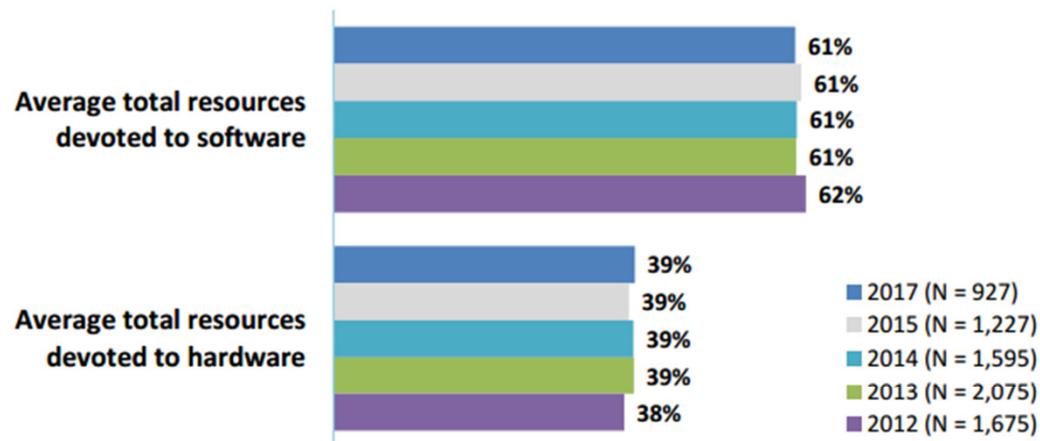


Dans le domaine de l'embarqué, le plus gros du « manpower » fait du développement logiciel au sens large.

Note: Outside vendors worked with is 2.7 on average.



What is your development team's ratio of total resources (including time/dollars/manpower) spent on software vs. hardware for your embedded projects?



Note:
 In 2017, respondents averaged working on 2.1 projects at the same time.
 In 2015, respondents averaged working on 2.1 projects at the same time.
 In 2014, respondents averaged working on 2.0 projects at the same time.

Applications et développement

Idée reçue numéro 2

De toute façon, l'embarqué, c'est que pour des applications militaires ...

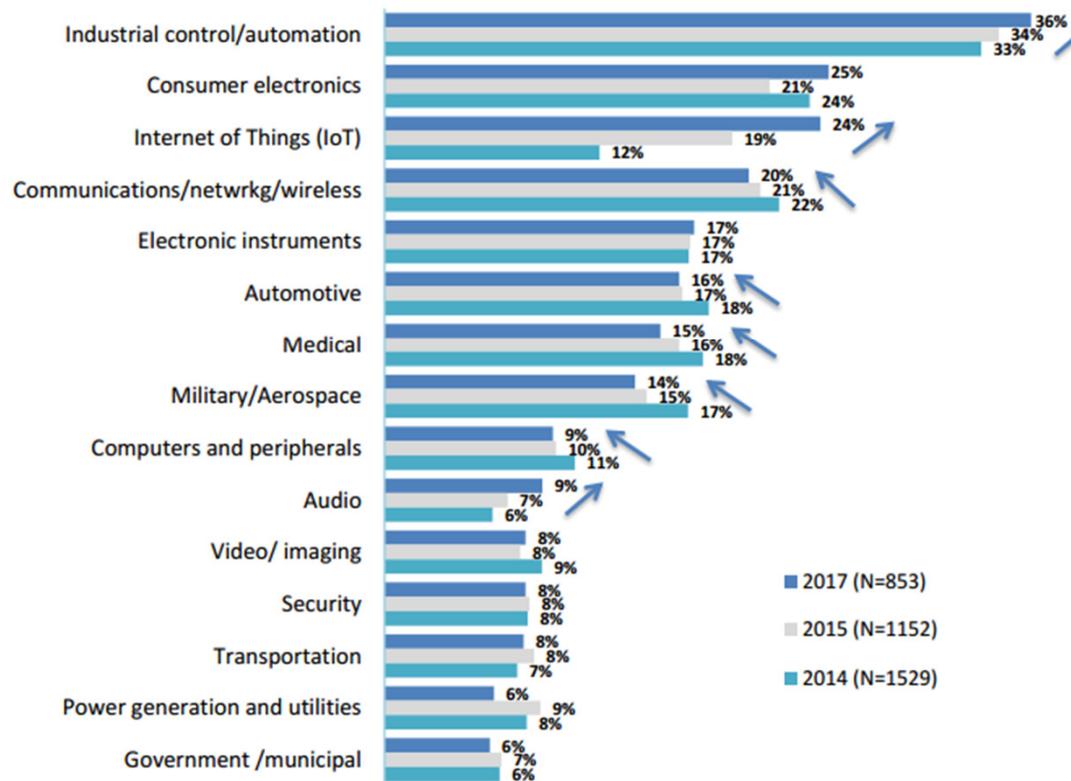
MR. GEEK





ASPENCORE

For what types of applications are your embedded projects developed?



Idée reçue numéro 3

Il faut apprendre Python et Java, le C ?
pfff c'est mort ! L'assembleur ... c'est
pour les musées !!!

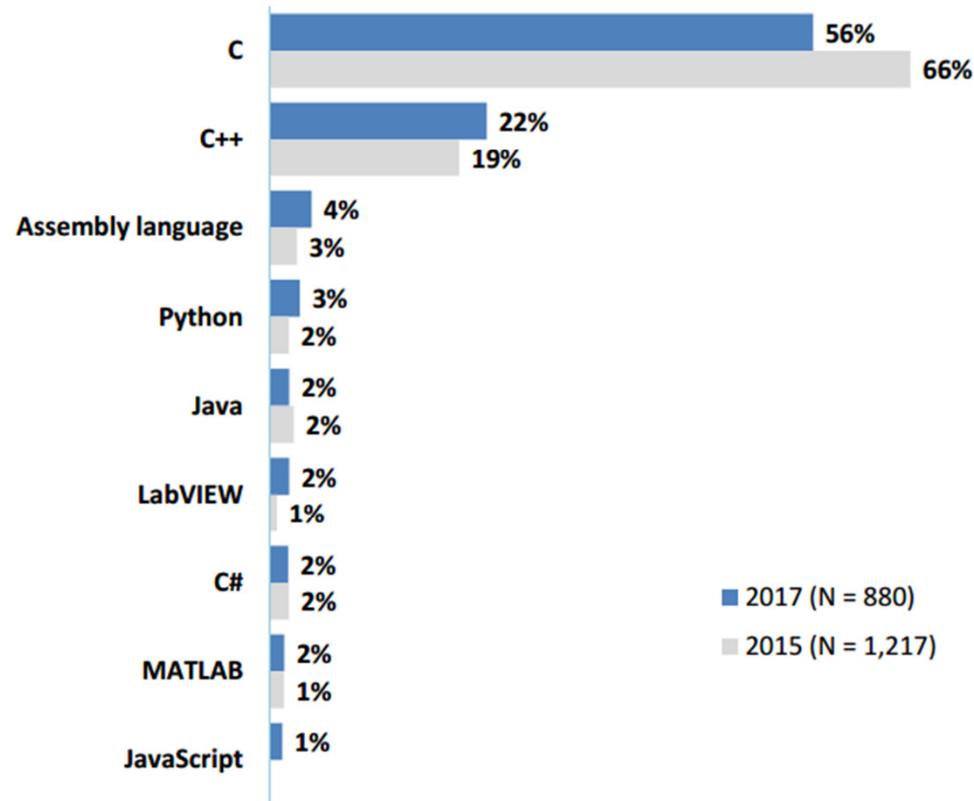
MR. GEEK



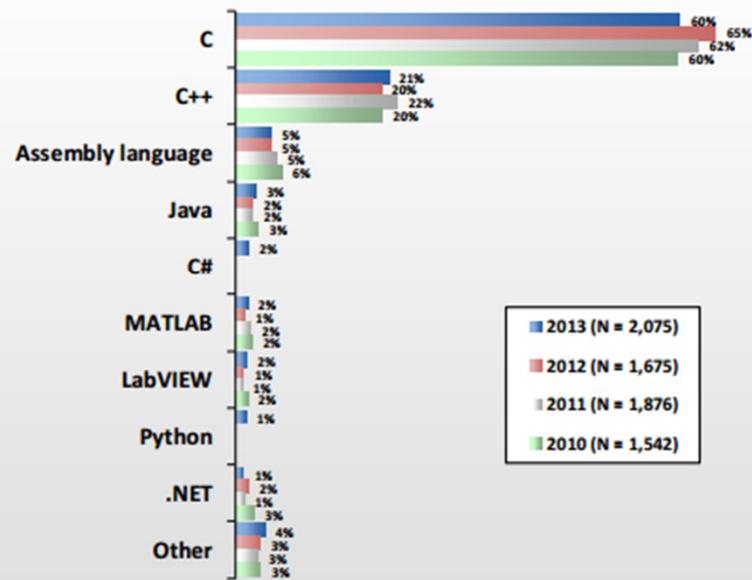


ASPENCORE

My current embedded project is programmed mostly in:



My current embedded project is programmed mostly in:



Note: C#, Python and Ada were added in 2013. Ada was under 1%.



Copyright © 2013 by UBM. All rights reserved.

Language(s) Used to Develop Software for Current Project

(Percent of Respondents)

Ada	4.3%	256
Ada-based SPARK	0.8%	256
Assembly	46.5%	256
C	85.9%	256
C++	42.2%	256
C#	6.2%	256
Forth	0.8%	256
Java	14.1%	256
LabVIEW	7.0%	256
MATLAB	16.8%	256
Perl	6.3%	256
Python	9.4%	256
SDL	0.8%	256
SystemC	2.3%	256
UML	9.0%	256
In-house developed	3.5%	256
Other	8.6%	256
Don't know	0.4%	256

Note: Percentages sum to over 100% due to multiple responses.

Language(s) Used to Describe IC/SoC Designs for Current Project

(Percent of Respondents)

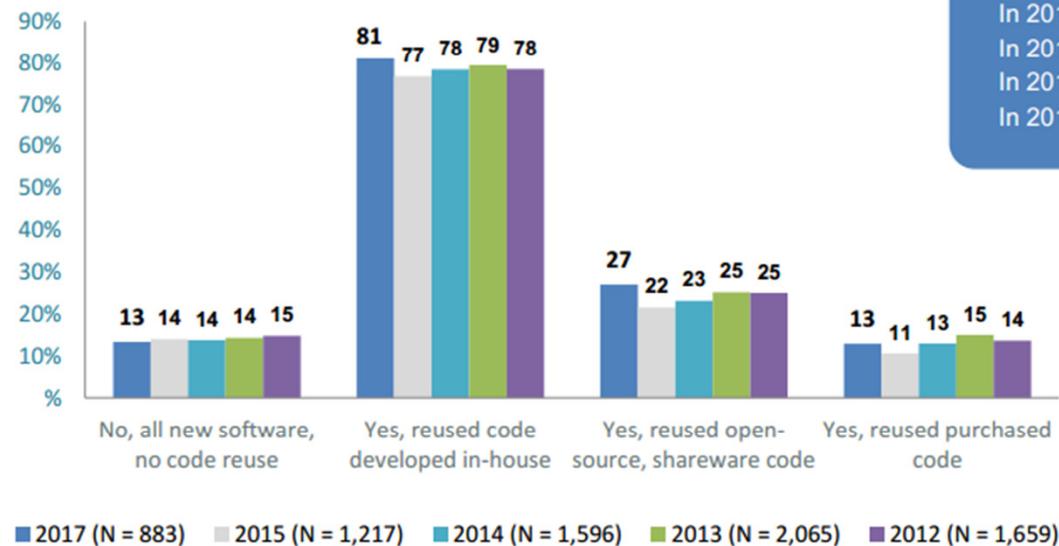
Netlists	8.3%	36
C	36.1%	36
C++	44.4%	36
e	5.6%	36
OpenVera	5.6%	36
PSL	2.8%	36
System Verilog	22.2%	36
SystemC	22.2%	36
Verilog	38.9%	36
VHDL	41.7%	36
In-house developed	5.6%	36
Other	0.0%	36
Don't know or N/A	11.1%	36

Note: Percentages sum to over 100% due to multiple responses.



ASPENCORE

Does your current project reuse code from a previous embedded project?



In 2017, 87% reused code.
 In 2015, 86% reused code.
 In 2014, 86% reused code.
 In 2013, 86% reused code.
 In 2012, 85% reused code.

Note 1. Multiple choice for "Yes" answers (a respondents can select more than one type of reused code).

Note 2. 76% of respondents also reused hardware or hardware IP.

EE|Times embedded

2017 Embedded Markets Study

© 2017 Copyright by AspenCore. All rights reserved.

Processus de développement

Idée reçue numéro 4

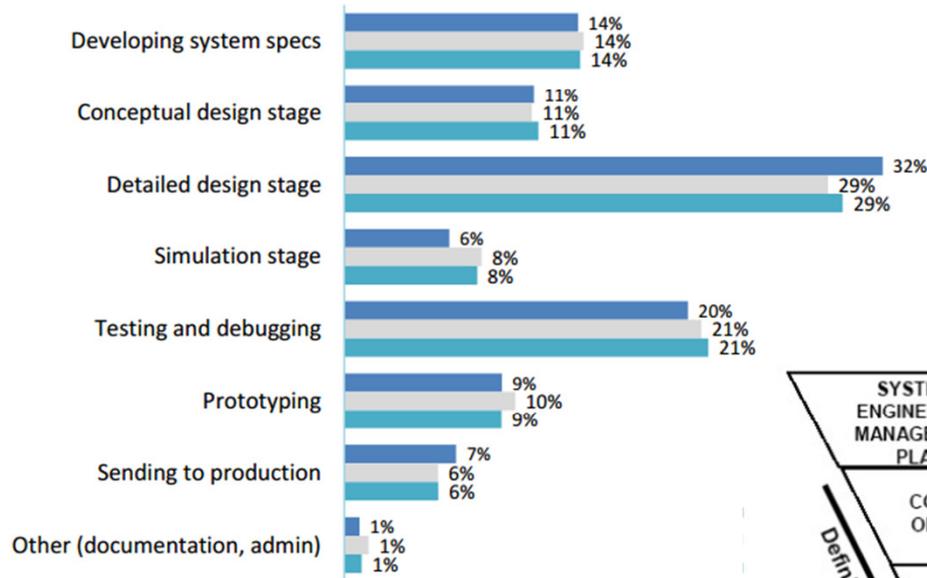
Le plus important dans le projet, est le codage, ça prend 90% du temps !

MR. GEEK



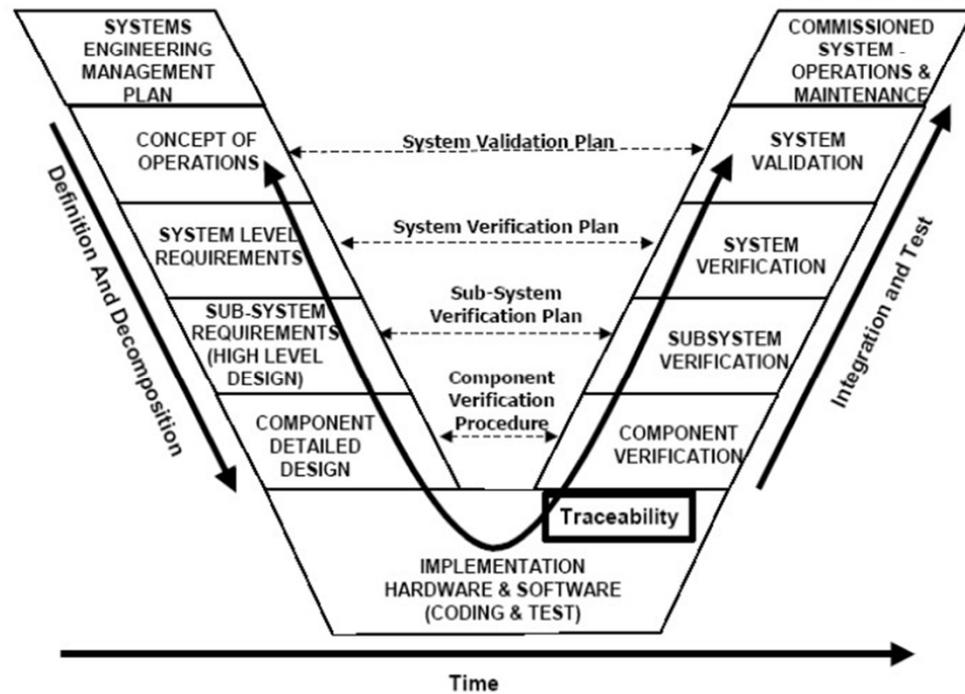


What percentage of your design time is spent on each of the following stages?



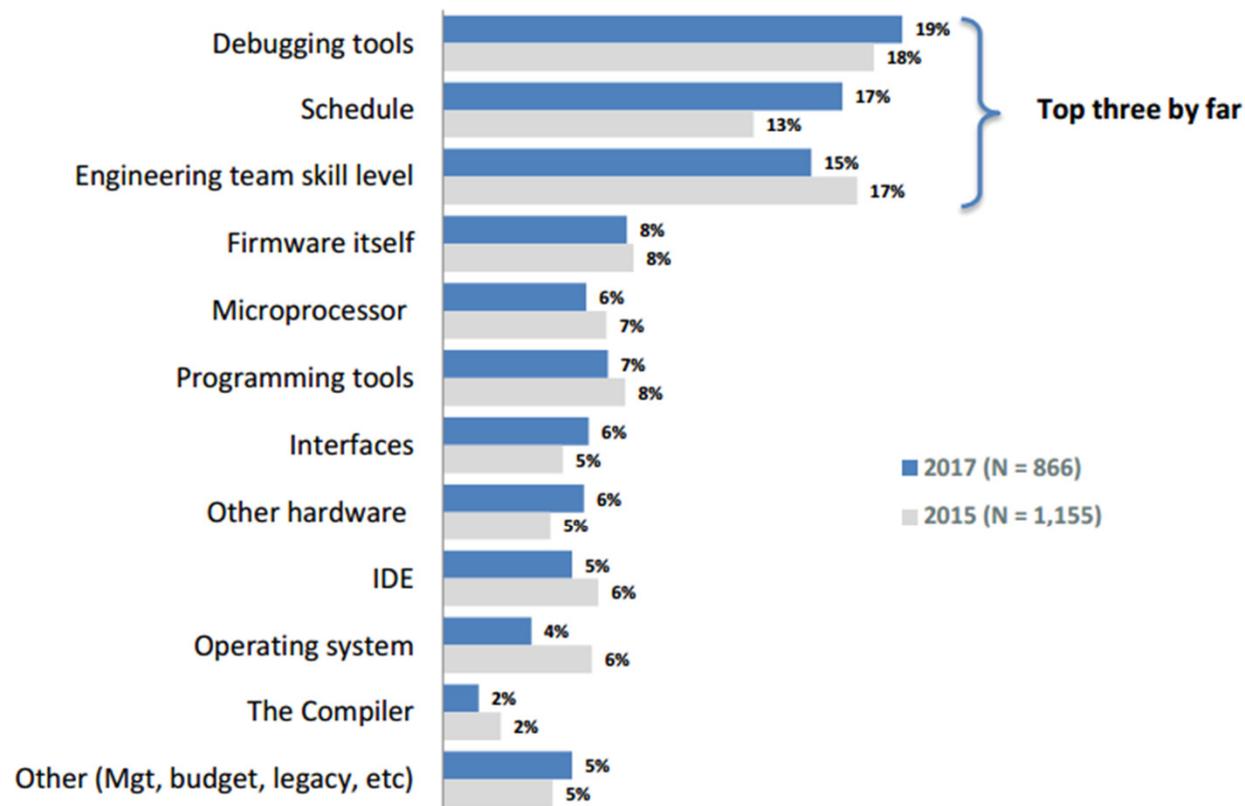
EE Times embedded

2017 Embedded Markets Study





If you could improve one thing about your embedded design activities, what would it be?



Idée reçue numéro 5

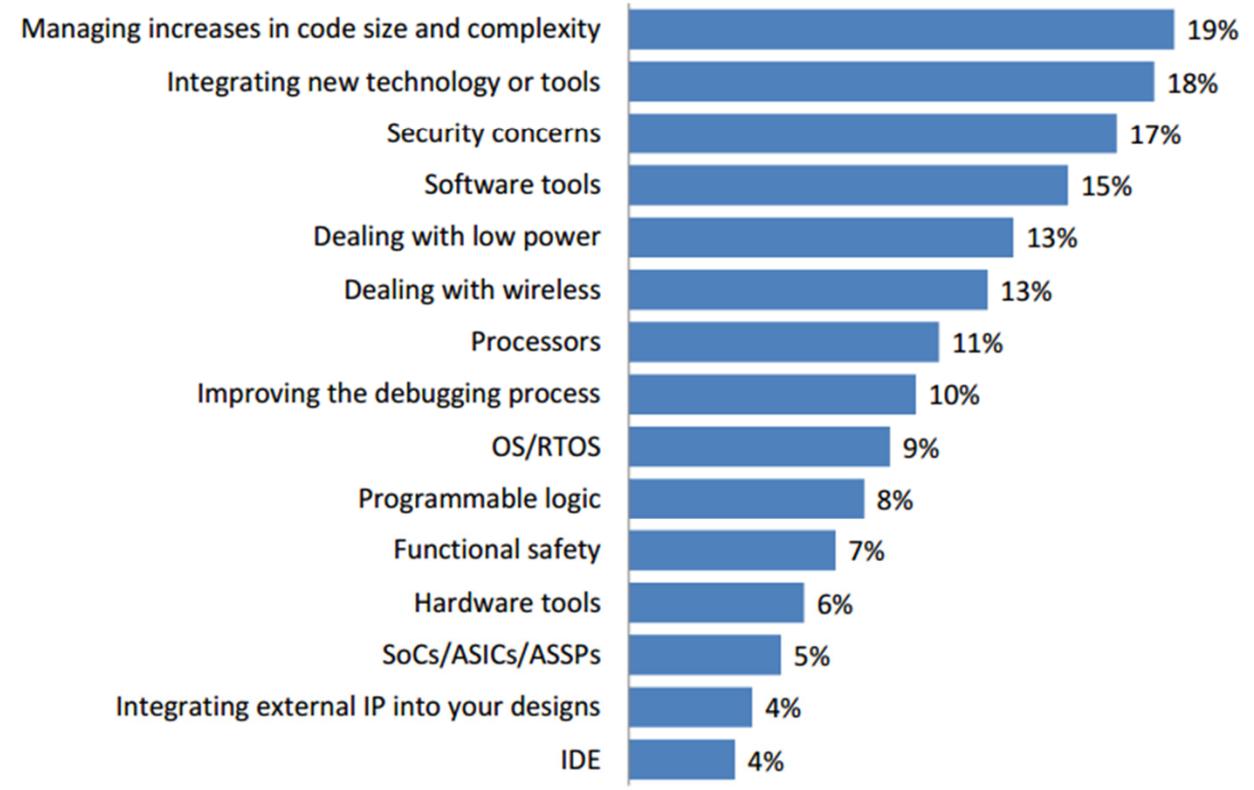
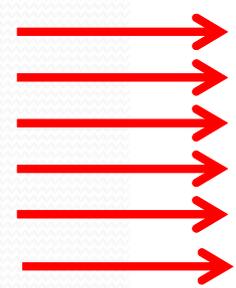
Les défis pour les systèmes embarqués
viennent des technos matérielles !

MR. GEEK

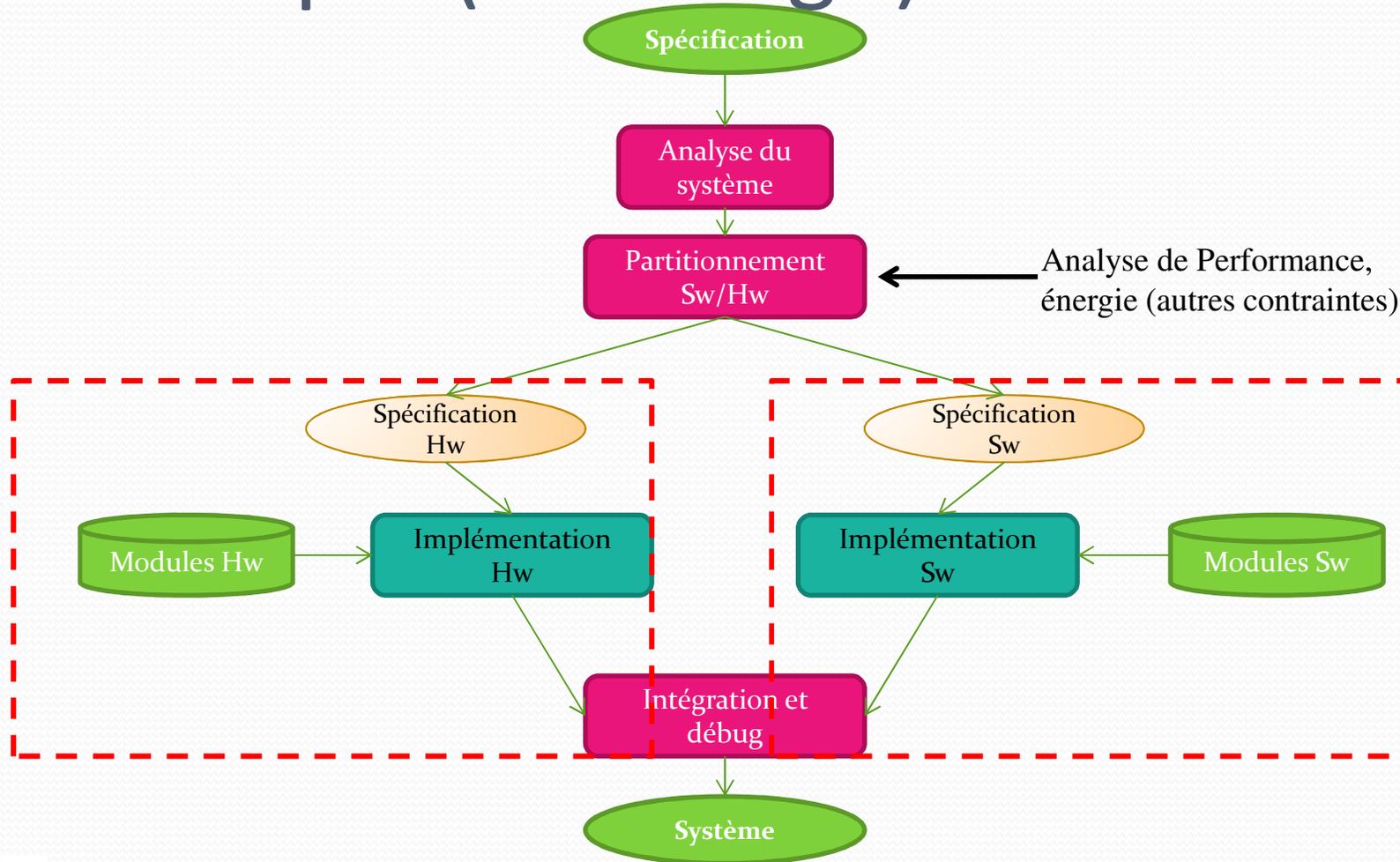




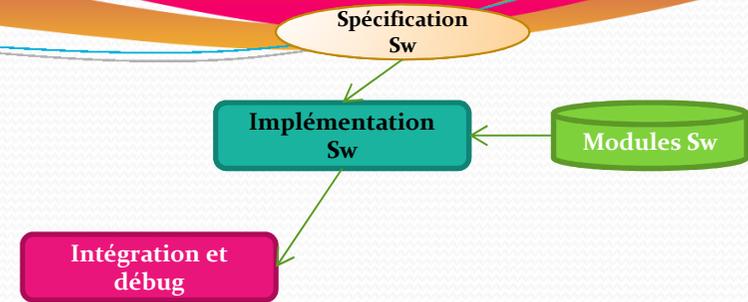
Thinking about the next year, what areas will be your greatest technology challenges?



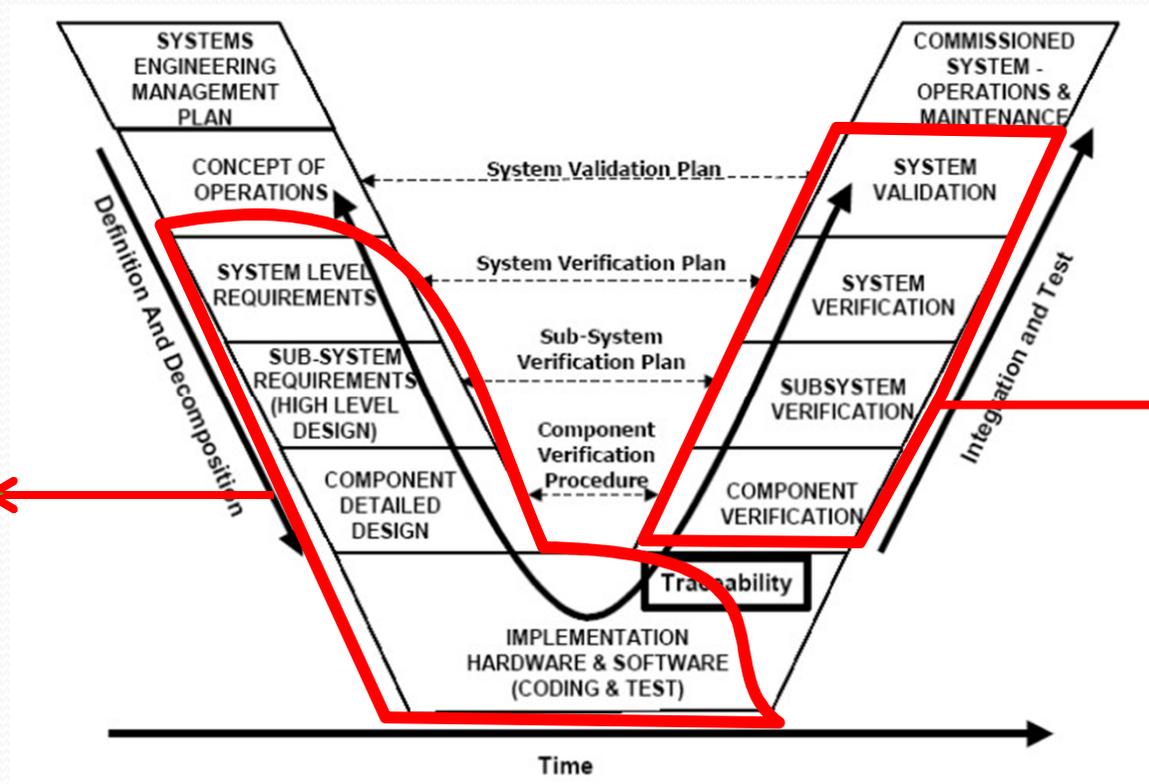
Flot de conception d'un sys. Embarqué (co-design)



Conception Sw.

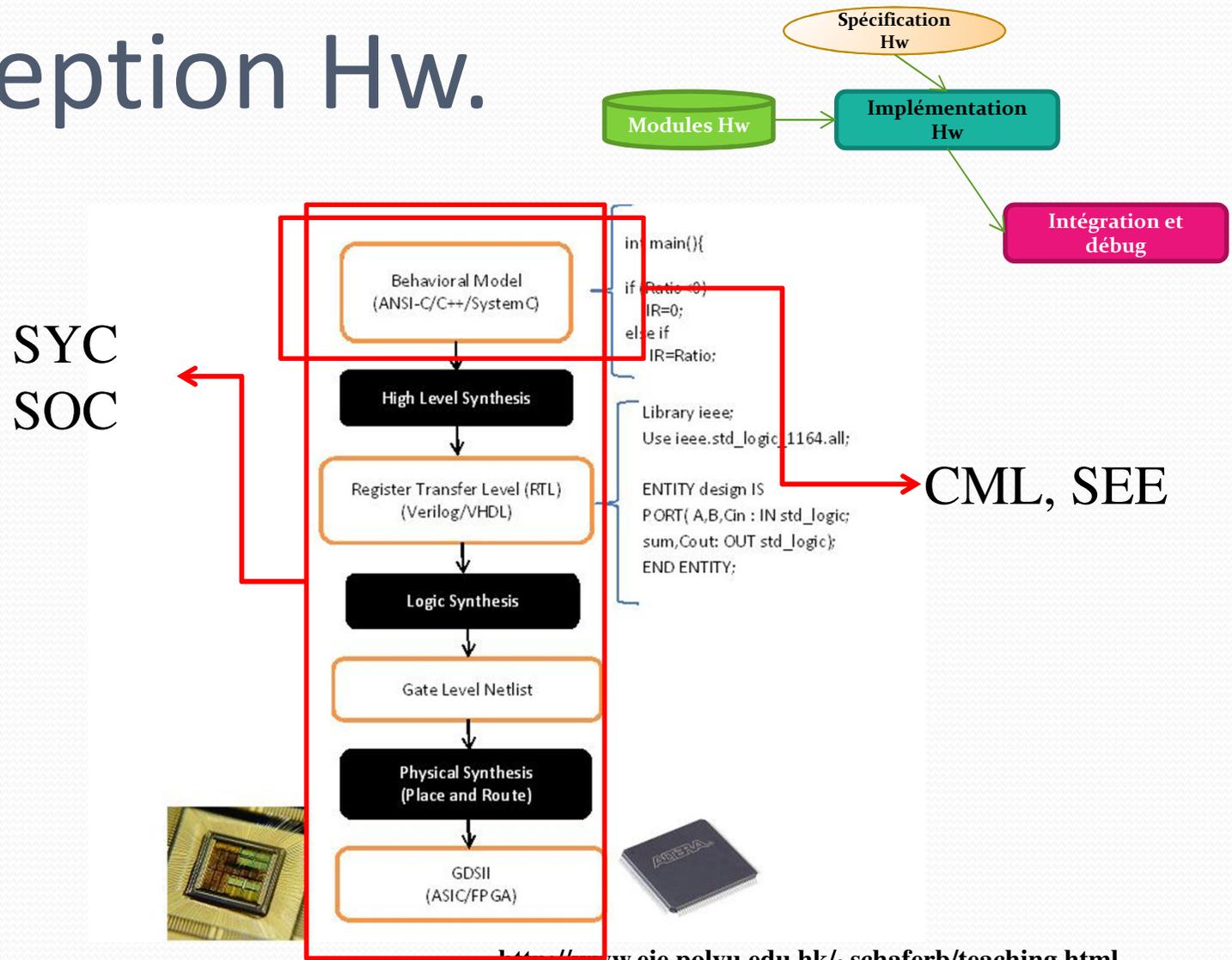


SOC
MOC
SEE
CML
SYC



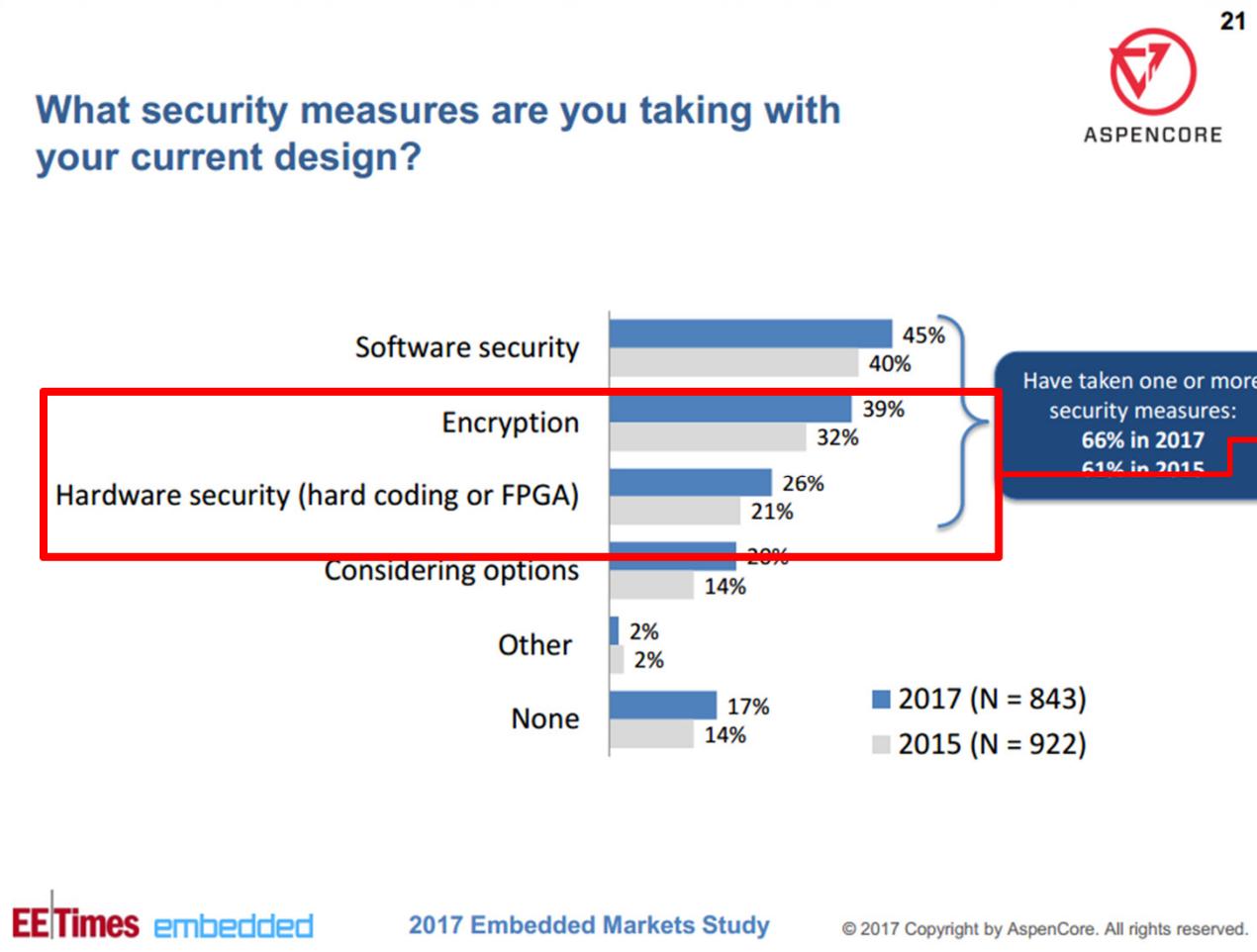
VFS

Conception Hw.



<http://www.eie.polyu.edu.hk/~schaferb/teaching.html>

Sécurité



Systemes d'exploitation

Idée reçue numéro 6

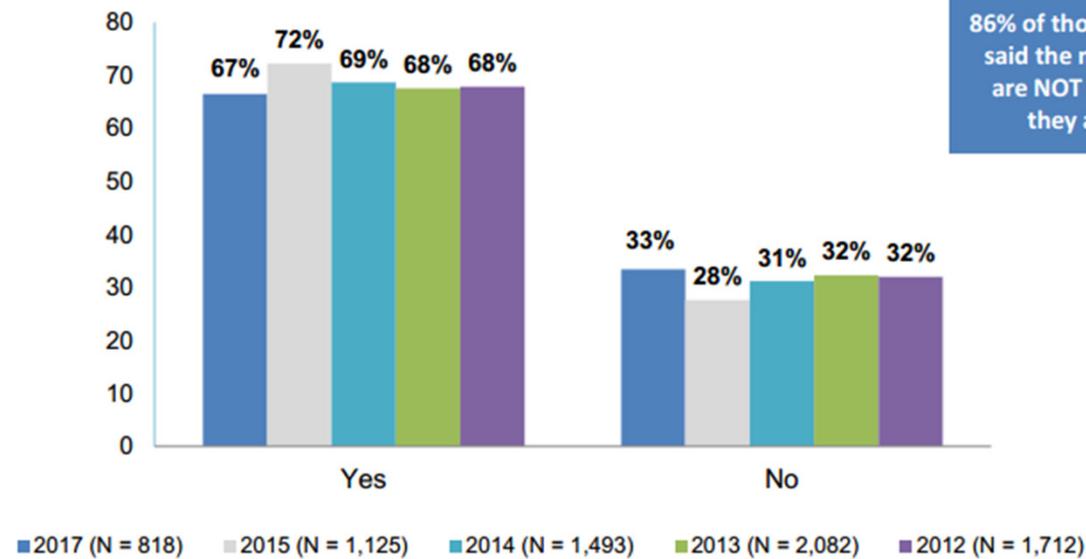
Les systèmes d'exploitation pour l'embarqué, c'est peu utilisé !

MR. GEEK



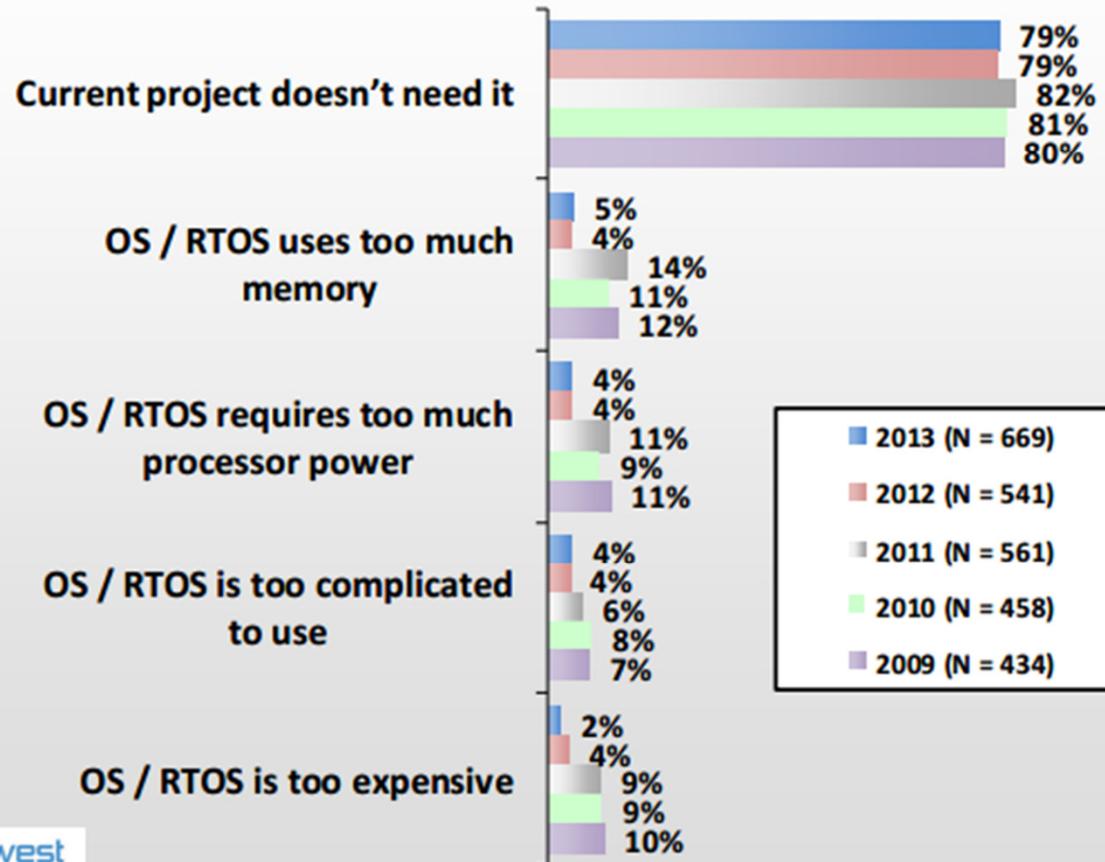
Does your current embedded project use an operating system, RTOS, kernel, software executive, or scheduler of any kind?

Fairly consistent usage of RTOS, kernels, execs, schedulers over past 5 years



86% of those not using RTOSes, said the main reason RTOSes are NOT used is simply that they are not needed.

If current embedded project does not use an operating system, RTOS, kernel, software executive, or scheduler of any kind, why not?



Copyright © 2013 by UBM. All rights reserved.

Which factors most influenced your decision to use a commercial operating system?

Top 19 reasons



Base = Those who currently use a "Commercial" OS/RTOS

Idée reçue numéro 7

Il doit y avoir peu d'entreprises développant des systèmes d'exploitation pour l'embarqué qui se partagent le marché dont **ANDROID!**, c'est sûr !

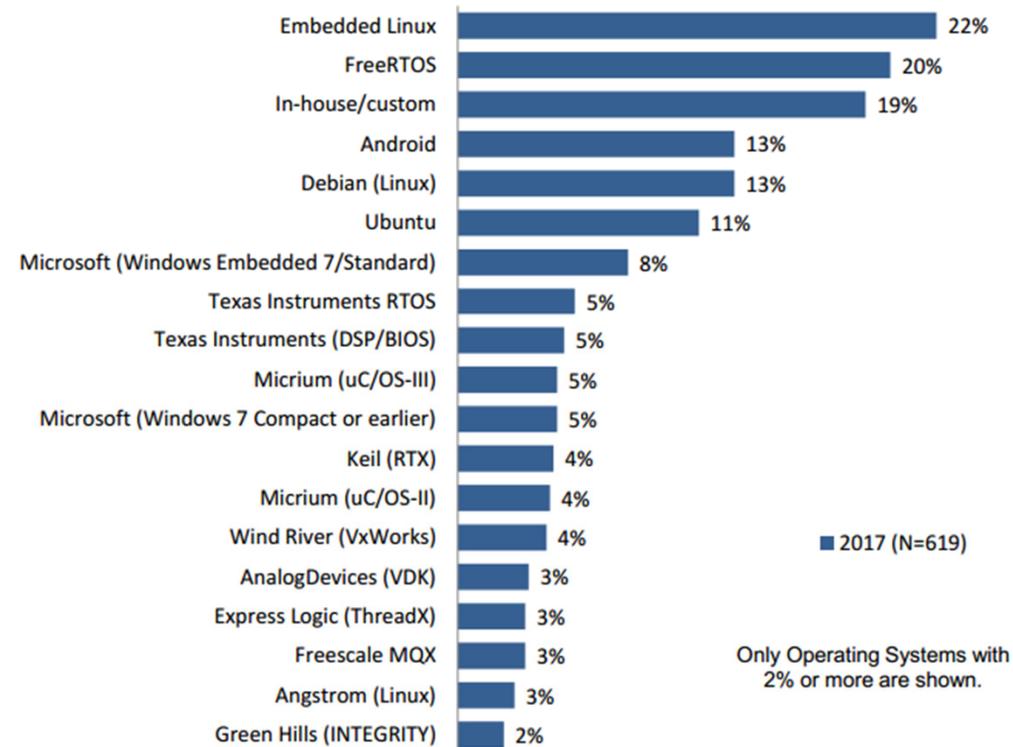
MR. GEEK





ASPENCORE

Please select ALL of the operating systems you are currently using.



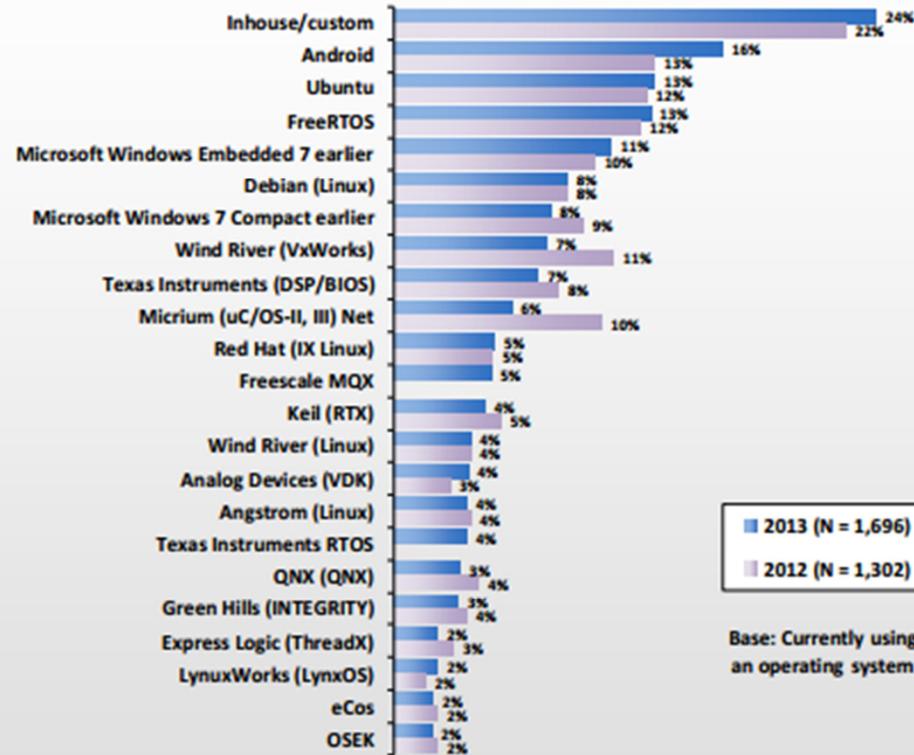
Base: Currently using an operating system

EETimes embedded

2017 Embedded Markets Study

© 2017 Copyright by AspenCore. All rights reserved.

Please select ALL of the operating systems you are currently using.



Base: Currently using an operating system

Only Operating Systems that had 2% or more are shown.



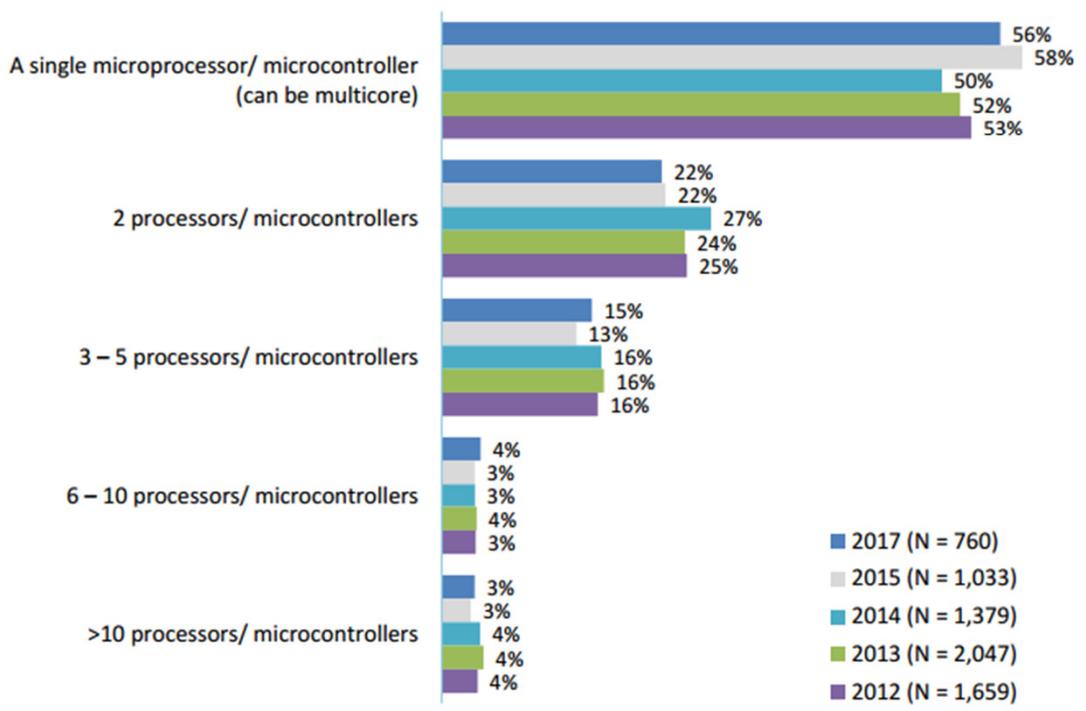
- OS embarqués → très utilisés, très critiques, aspects temps réel → **SEE & VFS**

Microprocesseurs



ASPENCORE

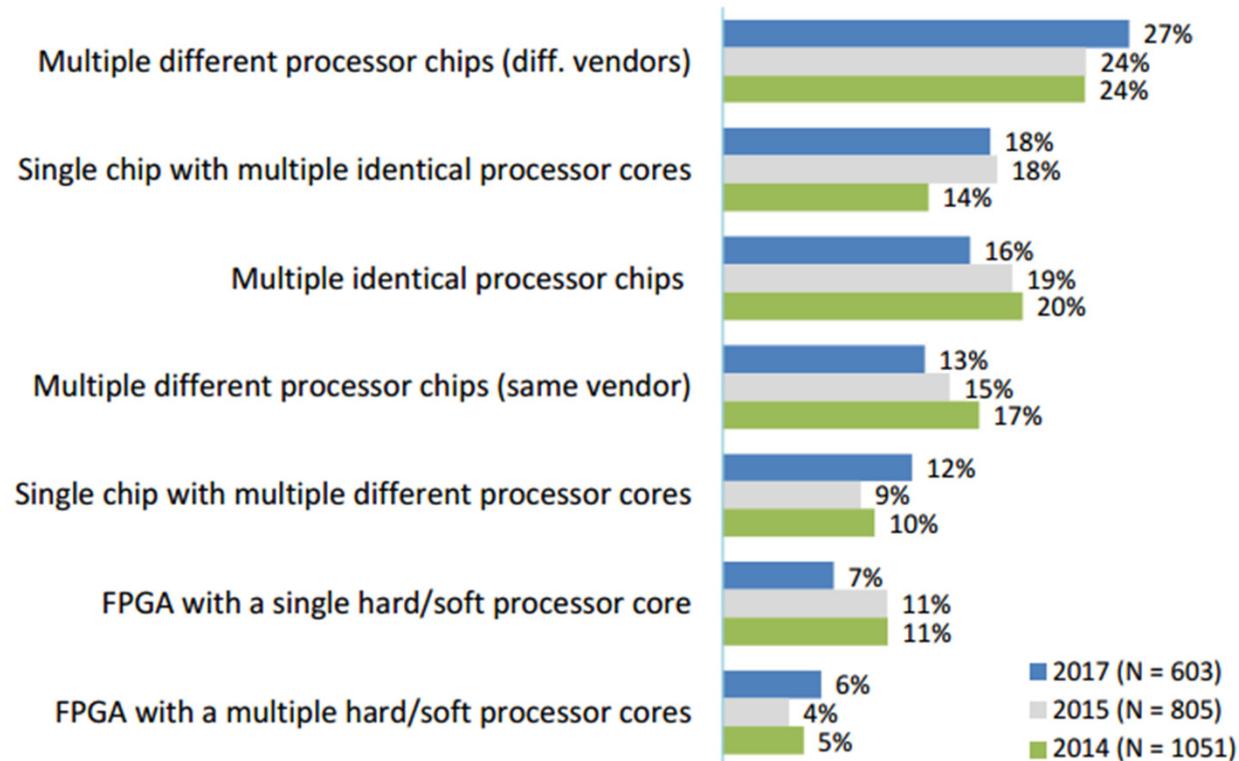
My current embedded project contains:



The average number microprocessor/micro controllers per project was:

- 2.3 in 2017
- 2.1 in 2015
- 2.4 in 2014
- 2.4 in 2013
- 2.3 in 2012

Does your embedded project contain...



Idée reçue numéro 8

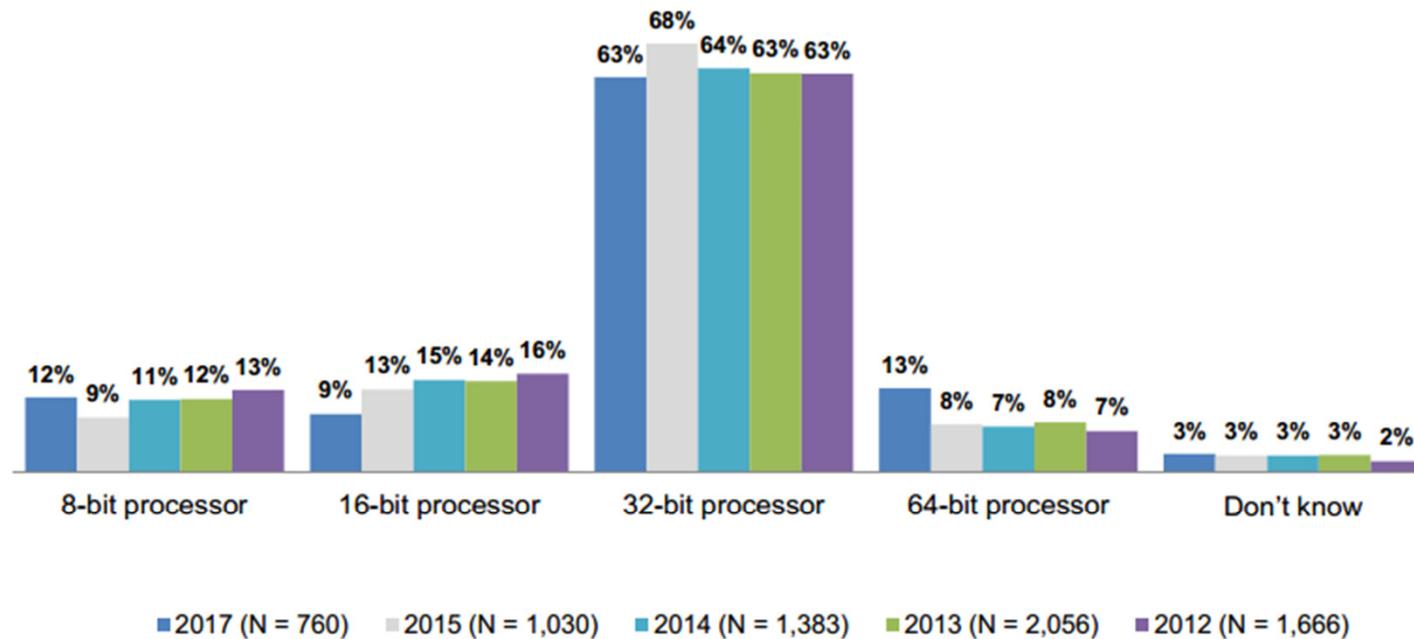
Les processeurs, ben c'est du 32bits, ou sinon du 64bits !

Quoi?, un processeur 8 bit, 16 bits ?? Il n'a pas mis à jour son cours celui là

MR. GEEK



My current embedded project's main processor is a:



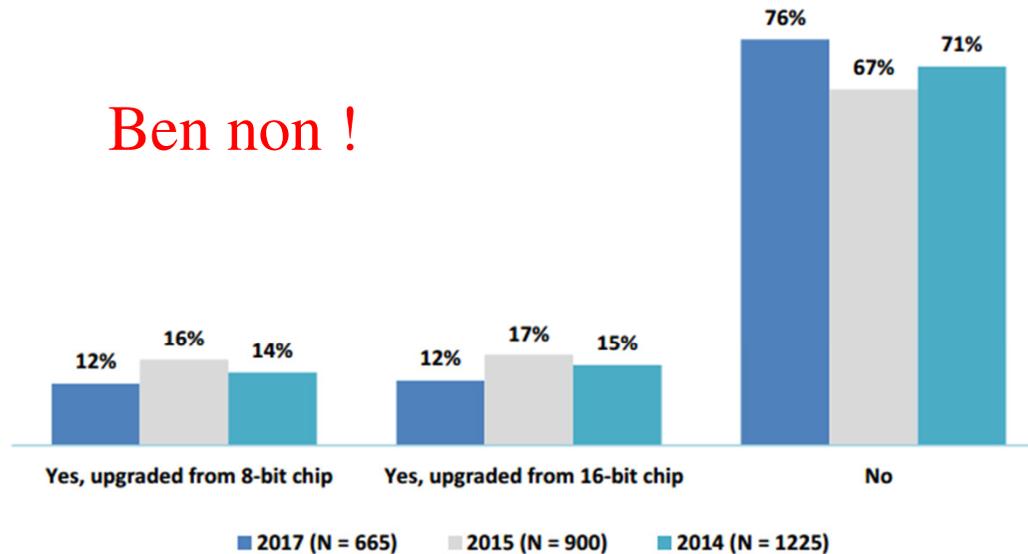
Note. 52% of respondents said additional processors (if any) were 32-bit processors, 18% said they added 8-bit processors, 14% added 16-bit processors, and 11% added 64-bit processors to their current embedded project.

Mais alors ! Ils vont migrer vers du 32 bits quand même !!!

Have you upgraded from an 8-bit or 16-bit chip to a 32-bit design in the last 12 months?



Ben non !



Idée reçue numéro 9

La fréquence d'horloge se mesure en GHz de nos jours !, en dessous c'est bon pour le placard ! Autant faire du calcul mental tiens!

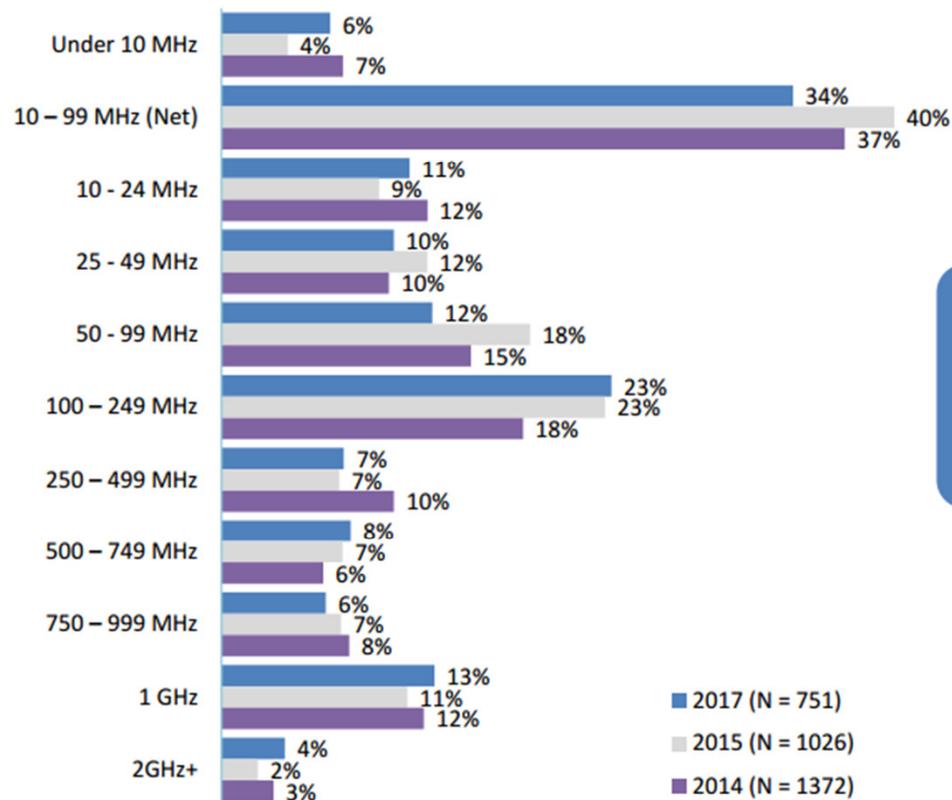
MR. GEEK





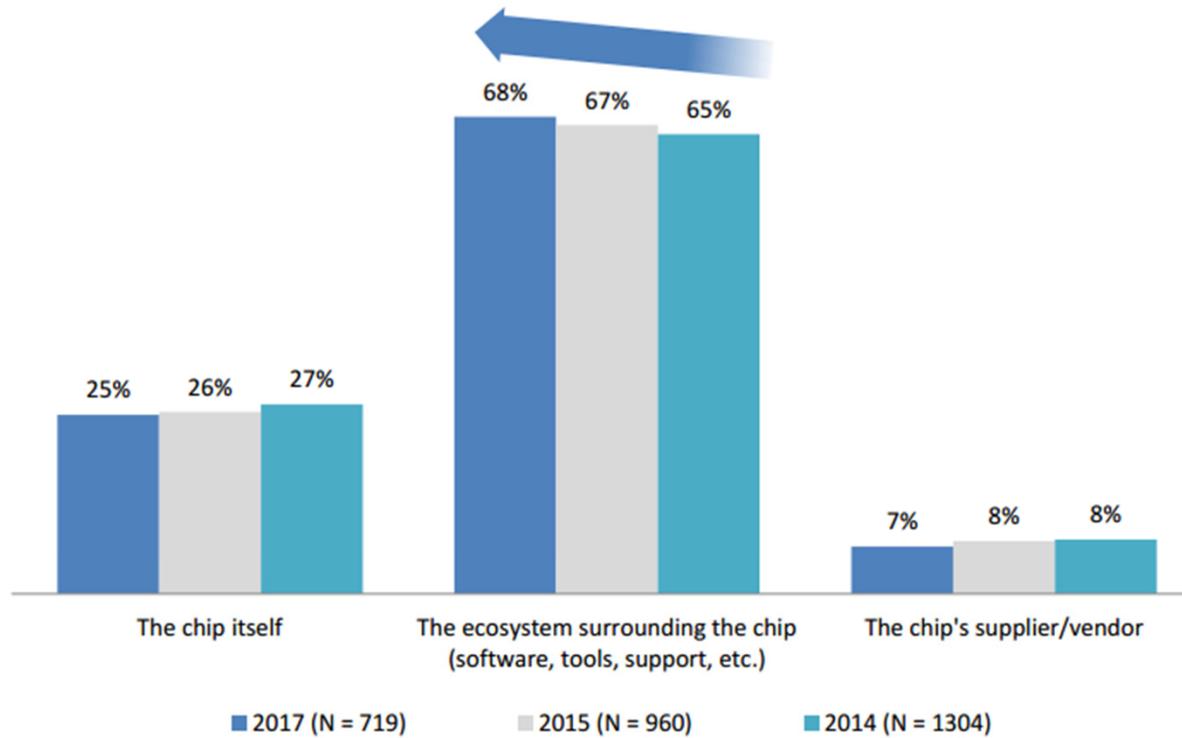
ASPENCORE

My current embedded project's main processor clock rate is:



The average processor clock rate was:
 445 MHz in 2017
 397 MHz in 2015
 428 MHz in 2014
 485 MHz in 2013

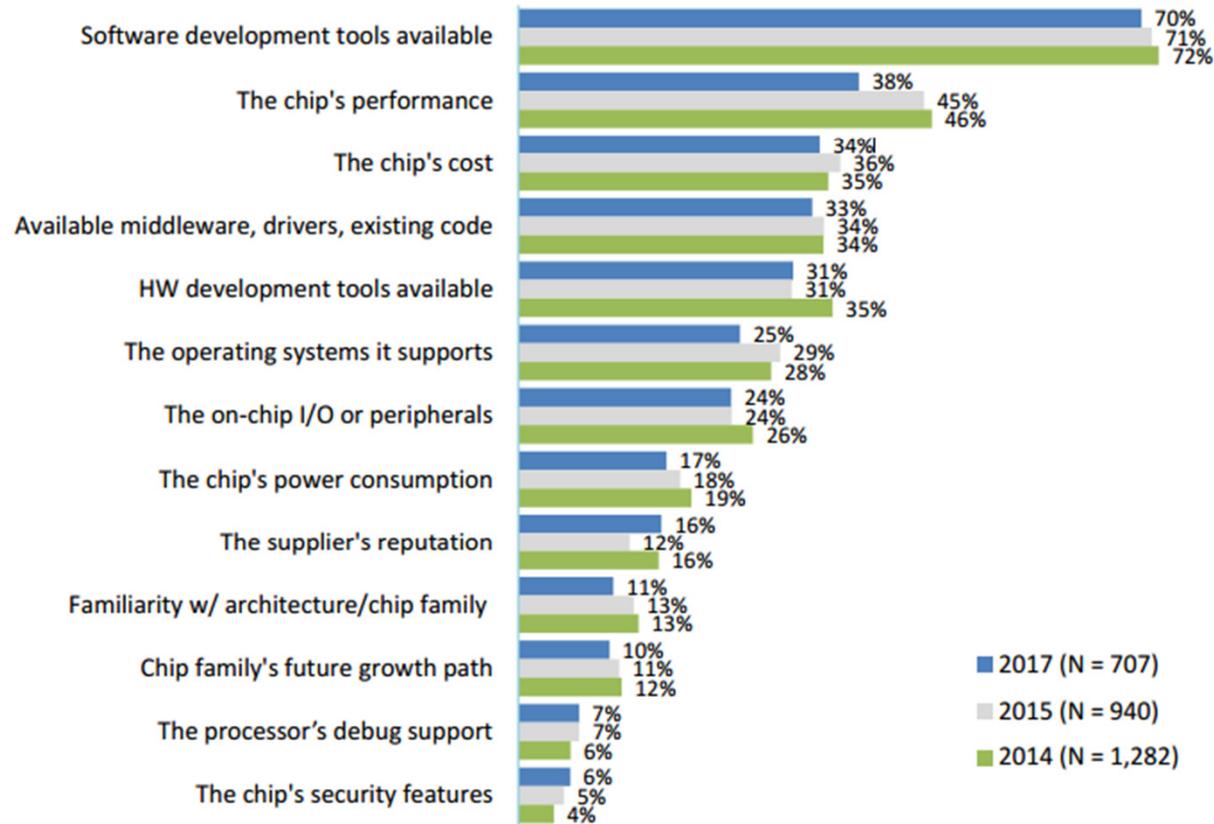
What's most important when choosing a microprocessor?





ASPENCORE

What are the most important factors in choosing a processor?



Processing Unit(s) Used on Current Designs
(Percent of Respondents)

MPU (Microprocessor Unit)	36.9%	279
MCU (Microcontroller Unit)	68.5%	279
FPGA (Field Programmable Gate Array) or PLD (programmable Logic Device)	31.2%	279
DSP (Digital Signal Processor)	26.9%	279
MPPA (massively Parallel Processor Array)	2.2%	279
SoC (System on Chip)	24.0%	279
Custom ASIC (Application Specific Integrated Circuit)	12.2%	279
System relies on external/distributed processing	9.7%	279
Other	0.7%	279

Note: Percentages sum to over 100% due to multiple responses.

Processing Architecture Used on the Current Project
(Percent of Respondents)

Single processor	61.2%	278
Multiprocessor (more than one processor on separate silicon)	21.9%	278
Multi-core (More than one processor core on the same silicon)	6.5%	278
Multi-core and multiprocessor	8.6%	278
Don't know	1.8%	278

Multiprocessing Methodology Employed for Current Project
(Percent of Respondents)

Asymmetric processing (application tasks/memory are processor specific)	57.4%	101
Symmetric processing (processors share application tasks/memory equally)	11.9%	101
Mix of both asymmetric and symmetric processing	12.9%	101
Don't know	17.8%	101

Multiprocessing Architecture Used in the Current Project
(Percent of Respondents)

Homogeneous cores/processors (processor are physically/architecturally similar)	29.7%	101
Heterogeneous cores/processors (processor are physically/architecturally different)	35.6%	101
Mix of both homogeneous and heterogeneous cores/processors	15.8%	101
Don't know	18.8%	101

Idée reçue numéro 10

Les FPGAs, tu en a déjà vu toi dans un appareil!, c'est jamais utilisé, pourquoi on nous bassine avec, déjà en M1 et maintenant en M2 !!!

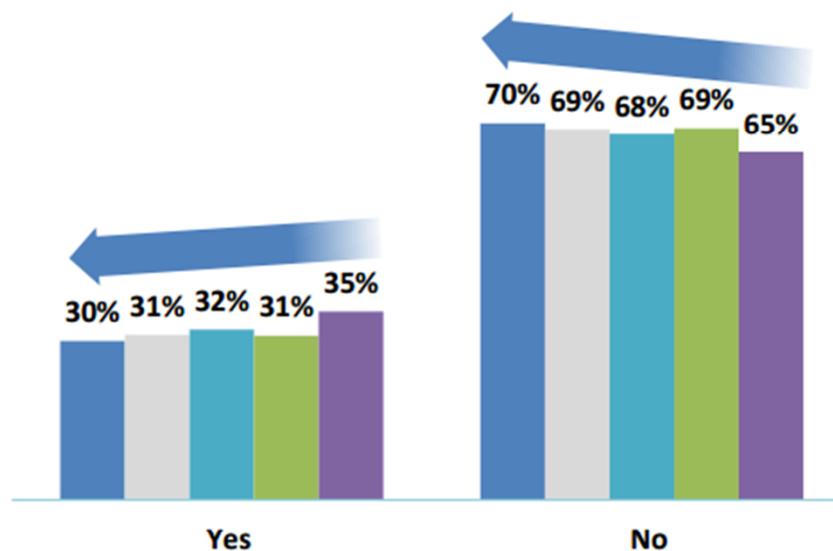
MR.GEEK





ASPENCORE

Does your current embedded project incorporate an FPGA chip?



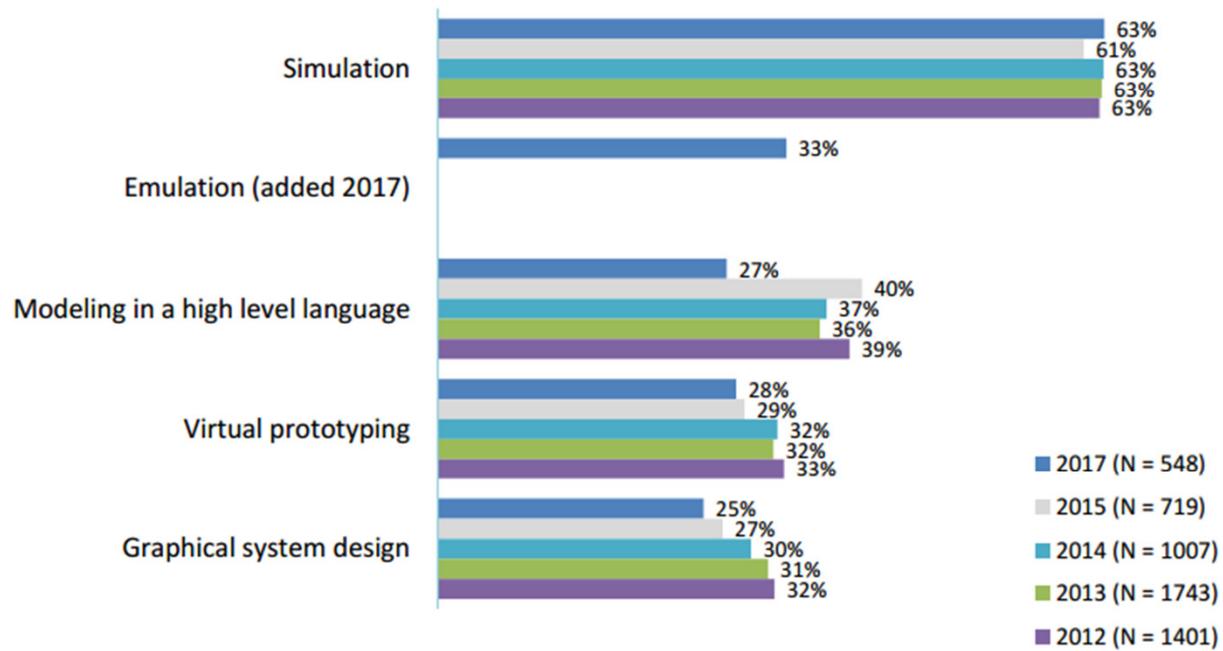
■ 2017 (N = 696) ■ 2015 (N = 959) ■ 2014 (N = 1,295) ■ 2013 (N = 2,073) ■ 2012 (N = 1,669)

Note 1: Among those not using FPGAs, only 12% said the trend towards FPGAs with built in multicore processors would change their mind, and 51% said "maybe" it would. And 37% said it would not change their mind.

Note 2: Only 25% of all respondents said they would use an FPGA in their **next** project further supporting the downward trend in using FPGAs. Those not using FPGAs in the future say they don't need the functionality, the cost of FPGAs is too high, or they consume too much power.



Which of the following design techniques will become more important to your designs in the future?



- Dans l'embarqué → plusieurs plateformes matérielles
→ l'outillage est crucial pour le choix du matériel.
 - Comment développer sur ces plateformes → CML, SOC, SYC
 - Comment développer/utiliser les outils → SYC, SOC

Le positionnement du master est il plus clair ?



Références

- Aspencore, « 2017 embedded market study », <http://m.eet.com/media/1246048/2017-embedded-market-study.pdf>
- UBM tech., « 2013 Embedded Market study », designwest, avr. 2013.
- VDC Research, « Embedded Software - 2009 market intelligence service », nov. 2009.
- M. Grünewald, « An introduction to embedded systems design » Univ. Paderborn.
- Frank Vahid and Tony Givargis, Embedded System Design: A Unified Hardware/Software Approach