Dr Simon Round, Head of Technology Management, MATLAB Conference 2015, Bern Switzerland, 9 June 2015

A Decade of Efficiency Gains

Leveraging modern development methods and the rising computational performance-price ratio

It’s all about Control

Imagine you are in a train

What are your expectations?

- Smooth ride in all weathers and seasons
- Quiet
- Getting to your destination on time, or
- Charging your smartphone

It’s all about Control

Imagine you are in a train

What are your expectations?

- Smooth ride in all weathers and seasons
- Quiet
- Getting to your destination on time, or
- Charging your smartphone

What are the consequences if the driving wheels are slipping?

It’s all about Control

Imagine you are in a train

What are your expectations?

- Smooth ride in all weathers and seasons
- Quiet
- Getting to your destination on time, or
- Charging your smartphone

What are the consequences if the driving wheels are slipping?

- Potentially expensive

Complex systems require sophisticated control

A Decade of Efficiency Gains

ABB's Journey

Let us prepare for a little journey

Looking at what ABB has been doing in the area of control, especially for driving trains

- We look back 10+ years
- Leveraging the rising computation power for the same price
- Leveraging modern development methods and tools

Delivering an expanded product portfolio with the same number of core development staff

Complexity is increasing

Systems/products are becoming more and more complex

- Using traditional development methods
  - would need extremely large teams
  - or a very long product development time

It is not just about reducing complexity or simplifying but

- Using the technological advances to our advantage
- Adapting to new development methods that reduce errors and make it easier to transfer knowledge

ABB
Business Unit Power Conversion

What do we do?

ABB is a global Power and Automation company

- In over 100 countries
- $40 billion revenue (2014)
- Formed in 1988 from merger of Swiss (BBC, 1891) and Swedish (ASEA, 1883) engineering companies

We help our customers

- to use electrical power efficiently
- to increase industrial productivity
- and to lower environmental impact in a sustainable way

Power and productivity for a better world

Power Protection

Solar Inverters

TOSA (Geneva)

Fast EV Chargers

ABB
Transportation

Product Group Transportation

Offers complete traction solutions, incl. transformer, traction converter and motor/generator

A traction converter is a product that contains all power conversion functions

- Controlling the motor speed
- Feeding braking energy back into the power network
- Providing wagon auxiliary power for lighting, heating & charging your phone
- Battery charging
- Traction/wheel-slip control

All operated by a single controller

Regional Trains

High-speed Trains

Traction Converter

Power Electronics Controller (PEC)

What is in a controller for a traction converter?

Power Electronics Controller

Converter controller

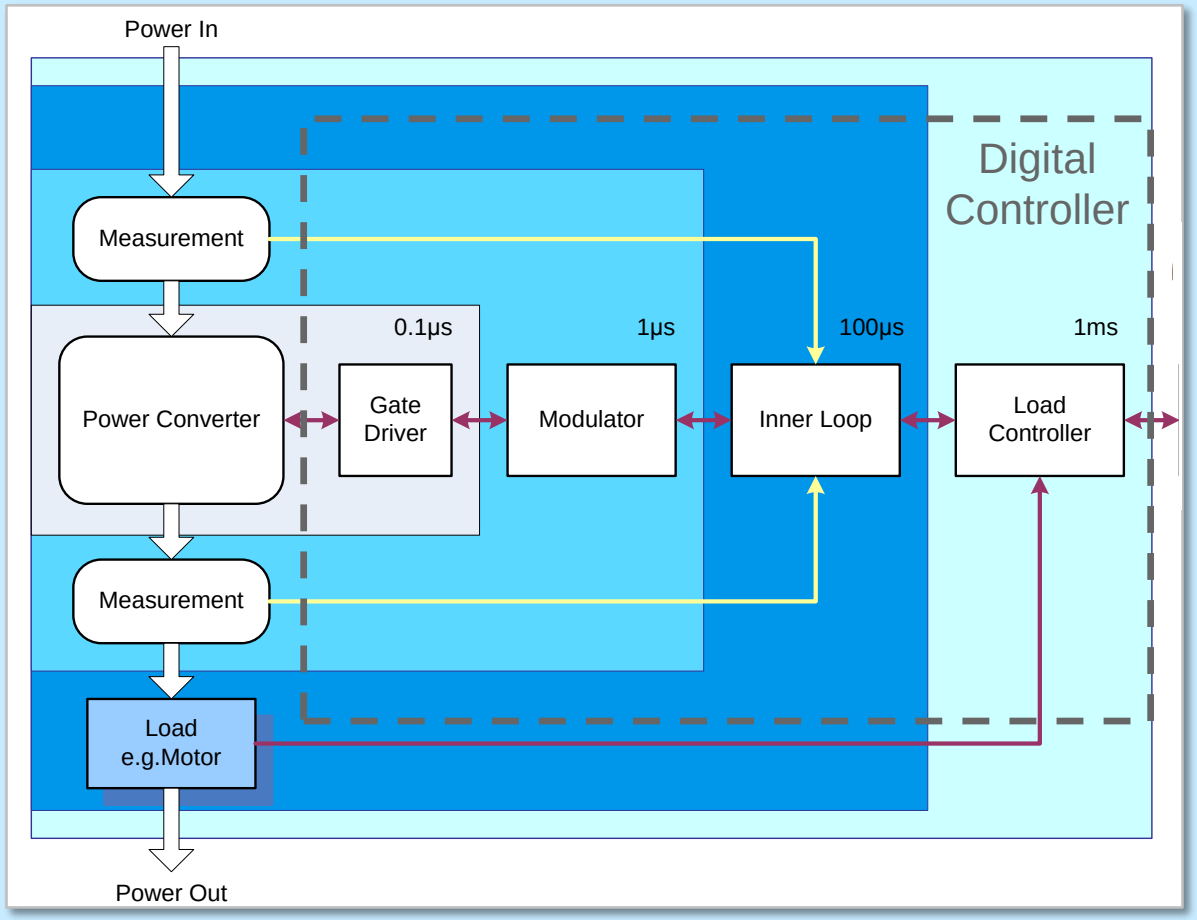
A controller is at the heart of the converter system

- Measure signals (speed, current, voltage, …)
- Calculates new output voltages to reach desired operating point
  - multi-rate, multi-tasking control system

It is a digital controller, which provides

- Repeatable operation
- Performance complex operations
- Adaptable to system changes

Digital control technology is changing rapidly



Digital controllers

Increasing computational performance-price ratio

Microprocessor evolution for industry

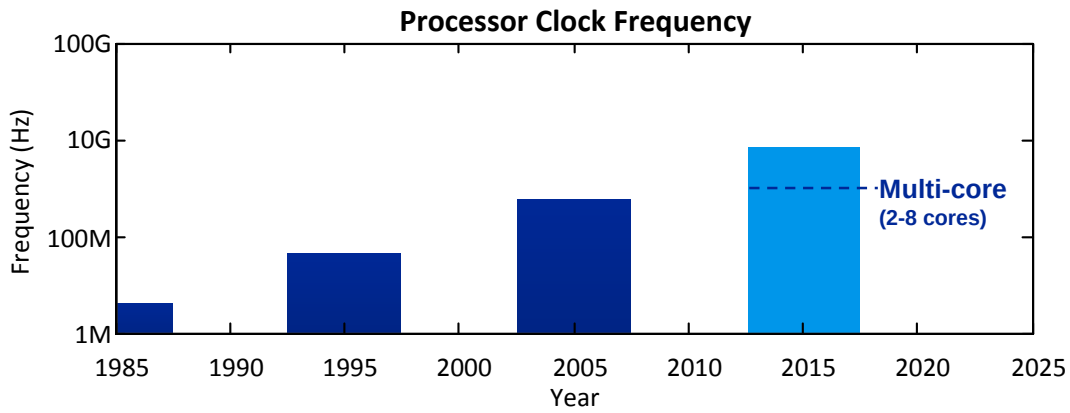
From the 1980s to today, industrial digital controllers have been taking the advantage of Moore's Law

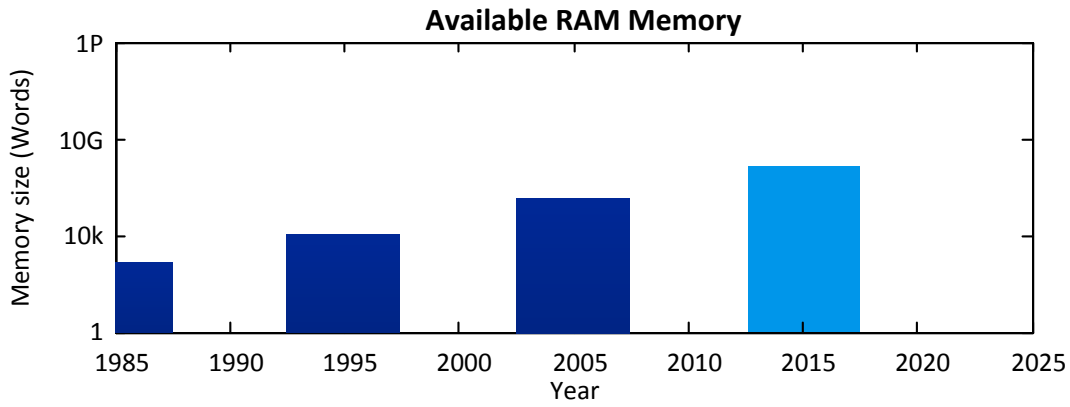
- More computation power, more memory capacity, more communication capability for lower prices
- Floating point computation is available in a $2 "chip"
  - makes control loop development easier

Industrial controllers today use multicore processors, programmable logic, and high-speed communications

  - and they can come as a single chip solution

More performance for less money…





ABB's Power Electronics Controller
AC 800PEC

ABB's digital controller

- First introduced into products in 2002
- Leveraged available high-performance, floating-point microcontrollers with programmable logic (FPGA)
- Was extremely expensive at the time
- Why pay more for the hardware?
  - Wanted control software development faster by
    - having fewer people interfaces
    - minimizing manual coding bugs

Power Electronics Controller today

- A dual-core Power PC/ARM CPU and large FPGA
  - Up to 1.2GHz
  - 0.5GB Memory – both DRAM and Flash
- Allows implementation of multiple control systems in one physical controller
- Over 30,000 controller units produced to-date
- Utilized in a large range of ABB products from trains to wind turbines to power stations

It is not just about the hardware!

Advances with digital controller software and development methods

Software development evolution

Traditional programming needs many steps

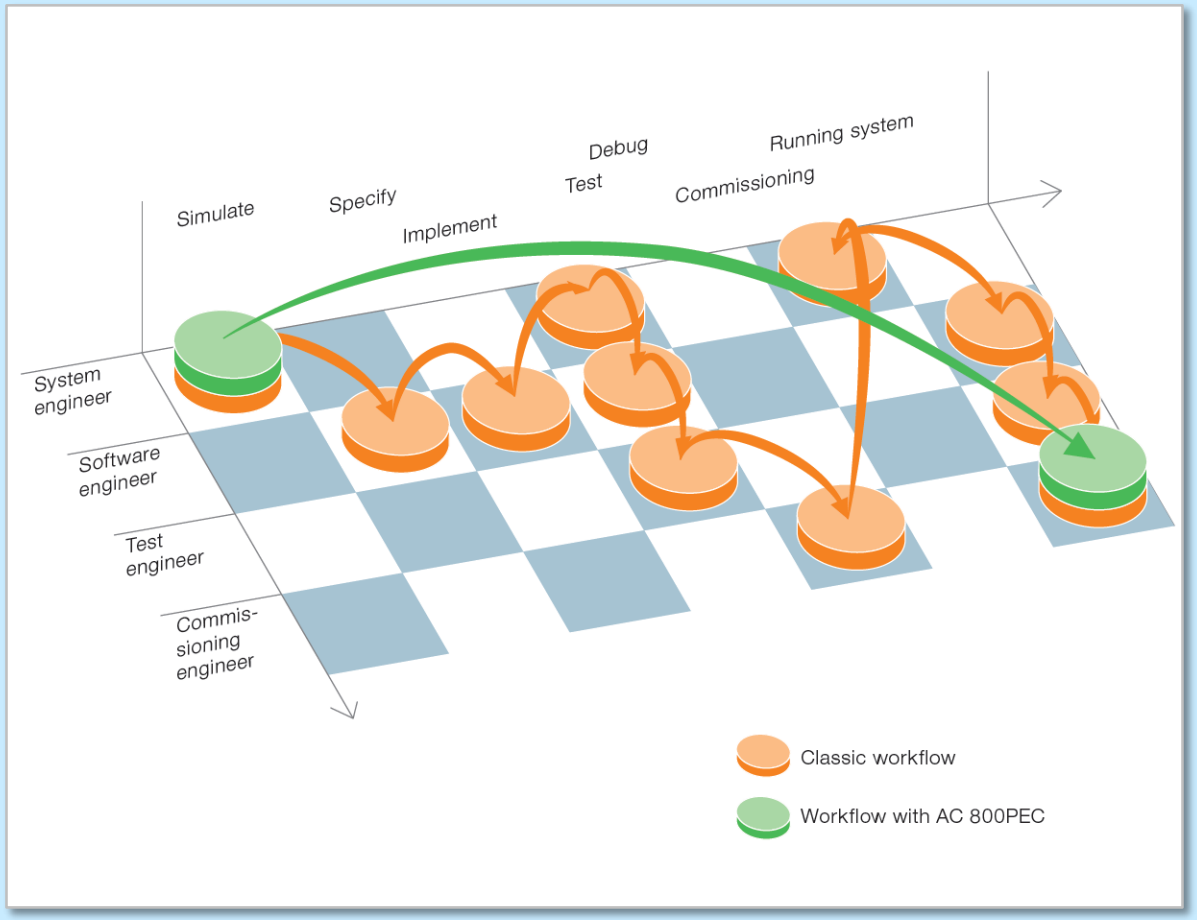
- 1980s Assembler programming – very low level
- 1990s C programming – higher level language

Leveraging increasing hardware performance to allow higher levels of software abstraction

- 2000s Graphical programming & code generation
- 2010s Model-Based Design – integrating design, simulation, code generation and testing

PEC uses MATLAB, Simulink & Stateflow to provide a Model-Based Design development

Do more with less steps



Model-Based Design and Automatic Code Generation

Bringing the development efficiency gains

Control engineer becomes empowered

A new way of developing

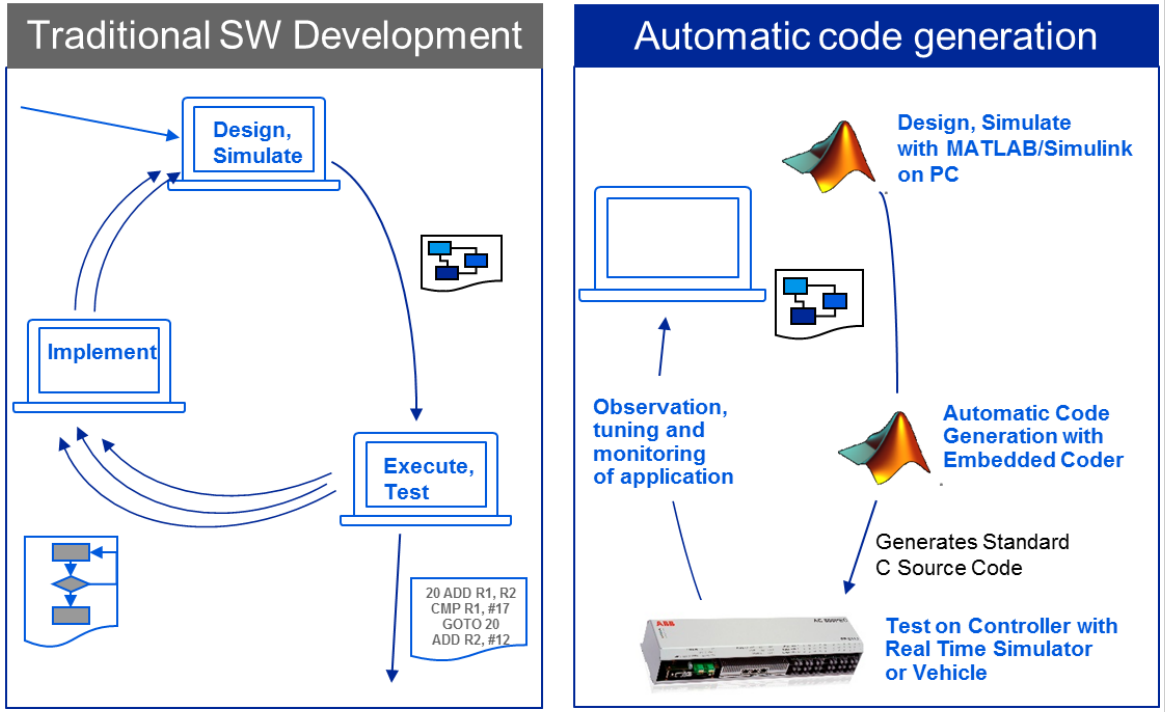
- Requirements → Models for controller and plant → Simulate & debug → Production code generation → Test

Deploy models to different targets

- Microcontroller, CPUs and programmable logic devices (FPGAs)

Used to spend 30% of time correcting the C code bugs

- Now we have to only fix control design/interaction issues in a simulation environment



Model-Based Design and Auto Code Generation

Bringing the development efficiency gains

Change in type of people needed

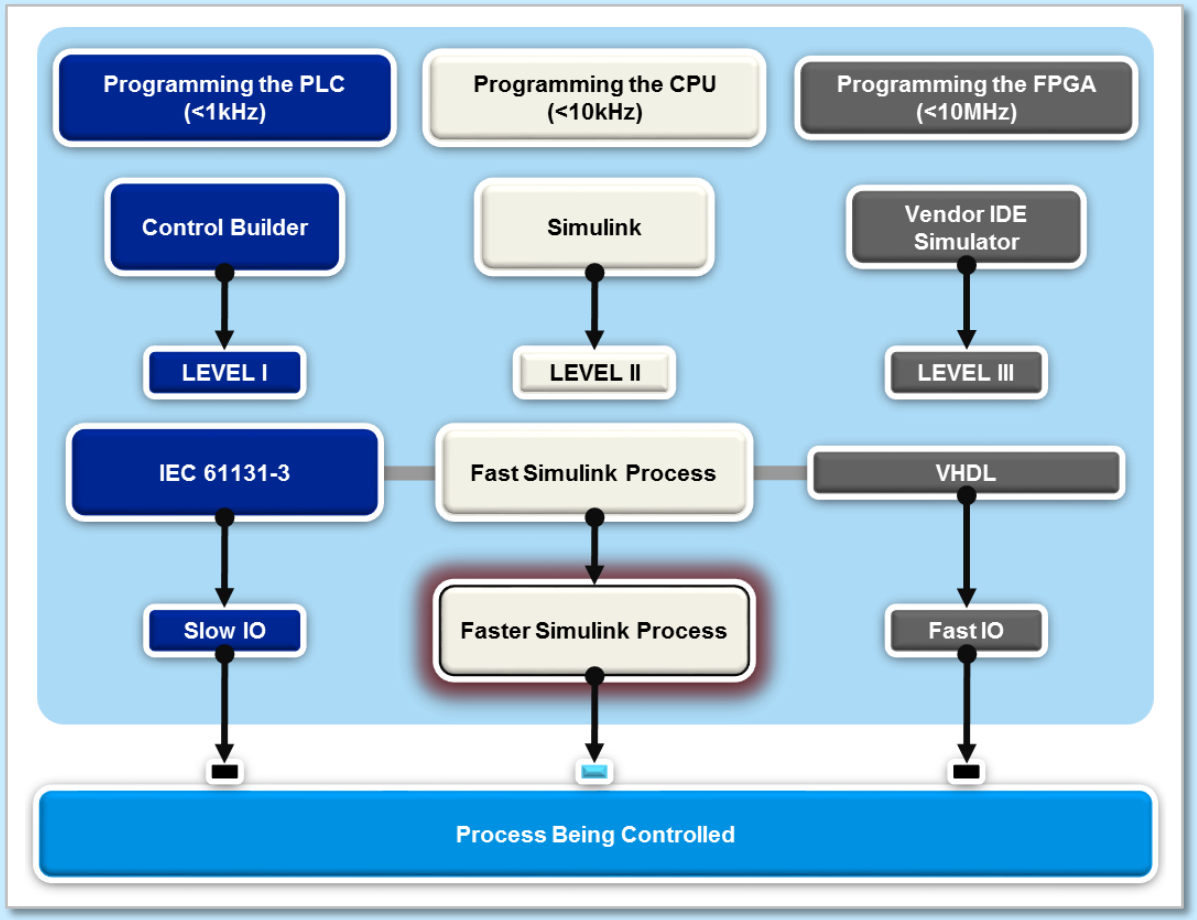
Do not have to have separate teams of control engineers and programmers

Do not have to find control engineers who can also program C and VHDL effectively

- They used to exist in the 80s & 90s

Been able to expand ABB's product portfolio and maintain it with smaller teams

- Plus having a faster time to market



Leveraging the computational performance-price ratio

Hardware advances enabled real-time performance of automatically generated code

A hard sell

People said C was not efficient as Assembler (too slow and too much memory used)

- Result: C won out since it was easier to develop and maintain.

People said that code generated from Model-Based Design is too slow and uses too much memory

- But today it is efficient as manual code
- Generate code for $2 floating point processors
- Much easier to develop, simulate (test) before deploying to the hardware

Other benefits

Easier to reuse models

Easier knowledge sharing

- Multiple team members can instantly understand the functionality from the graphical representation

Forms part of the documentation

Functionality that you simulate on your PC is the same as that runs on the controller

Model-Based Design for Traction Converters

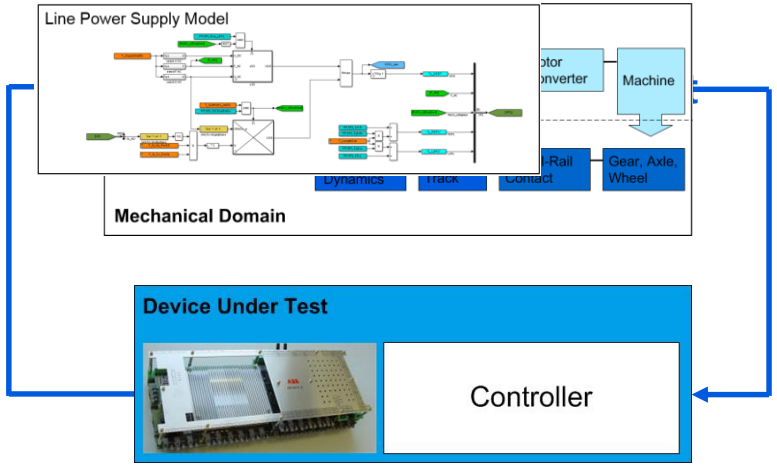
An example

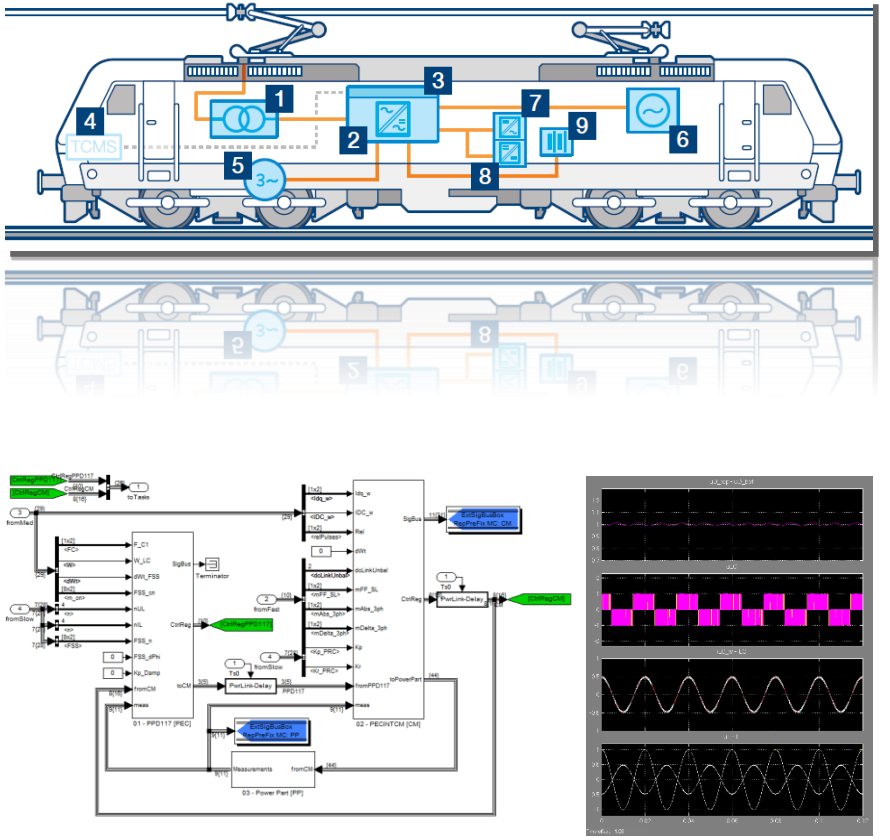
Where is Model-Based Design used?

From control of battery chargers, auxiliary power systems, to motor speed and wheel-slip control

- Large scale models with over 50,000 blocks

Simulation of the plant models e.g. motors, train dynamics





Not all is rosy as it seems!

New challenges brought about by Model-Based Design

Development methods

Model-Based Design is still software development but with graphical models!

Need to apply software engineering techniques

- In past everyone had their own graphical style
  - need guidelines and automatic checking
- Unit testing of functional blocks/subsystems
- Verification and validation
  - need to leverage automatic testing
  - use of Hardware-In-the-Loop to test before you go into the field

Working together and protecting knowledge

Protection of IP

- All your ideas are in the models
- How to share models with others without revealing all your knowledge?

Working in teams

- A single person developed the system when control was simpler
- Now systems are doing much more - need a team
- Support multiple people working simultaneously

Still need traditional software development

- Not all can be developed with Model-Based Design (yet) e.g. communication protocols, device drivers

Where is the future taking us?
Outlook for the next 10 years

Hardware and Software

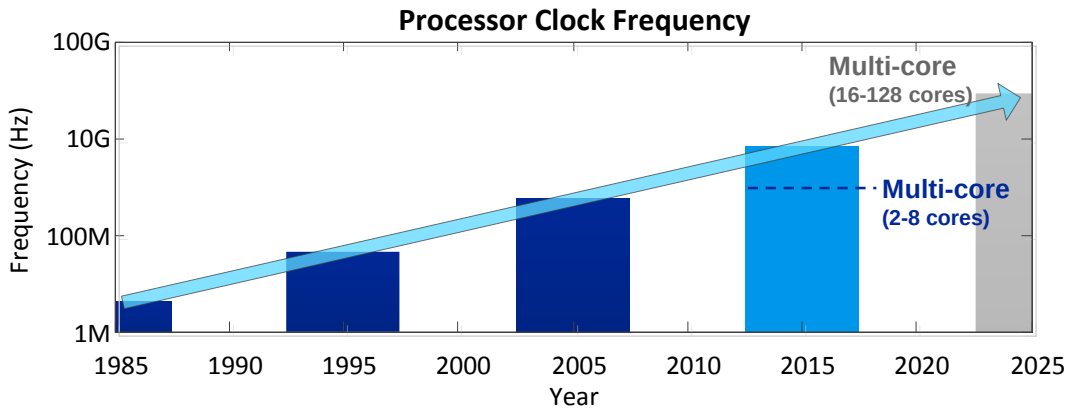
Processing Hardware

- Will get more performance in a single chip for same or slightly more cost
- More integration
  - even more functions on the same piece of silicon (cores, FPGAs, GPUs, communications)

Software

- Code generation for heterogeneous systems
- Late deployment to a mixture of cores, CPUs and FPGAs

Must utilize the power of the latest technologies



Summary
A decade of efficiency gains

ABB gains

10+ years of Model-Based Design experience

- MATLAB/Simulink and Automatic Code Generation

A lot of things learnt along the way

- Efficiency gains by getting new products to market faster
- Expanded product range by 4 times, supported by the same number of platform engineers
- Automatic code generation eliminates 30-50% of bugs introduced in manual coding

Things to remember

Treat it the same as traditional software development

- Need the correct processes in place even though it is graphically based
- Modelling is complex and must be done in the correct way
- Tools alone will not give you efficiency gains

Model-Based Design coupled with the modern controllers allows faster development

- Even for the low-cost $2 processors!

Complexity is here. Control it. Reduce to simpler steps, adapt ways of working, and leverage the technological changes in computational performance and tools

Power and productivity
for a better world™