In the Days of IoT Dealing with Software Parallelization for Heterogeneous Multicore Architectures

Yankin Tanurhan
Vice President R&D, Solutions Group

MPSoC, July 2014

Accelerating Innovation

THE INTERNET OF THINGS

There will be 25 billion devices connected to the Internet by 2015 and 50 billion by 2020.

Source: Cisco Systems Report: The Internet of Things – How the Next Evolution of the Internet Is Changing Everything.

Gigabyte
10^9
Terabyte
10^{12}
Petabyte
10^{15}
Exabyte
10^{18}
Zettabyte
10^{21}
Yottabyte
10^{24}

MORE DATA

Create	Store	Transport	Manipulate	Access	Secure

Source: Cisco Global Cloud Index (Oct 2013), Cisco Virtual Networking Index (May 2013), Wikipedia, BBC Future (Jun 2013)

IN A MINUTE

2 million searches	**571** new sites created	**72 hours** of video uploaded	**350 gigabytes** of data	**204 million** emails sent	**104 thousand** photos shared	**11 thousand** professional searches	**278 thousand** tweets
Google		YouTube	facebook		snapchat	LinkedIn	twitter

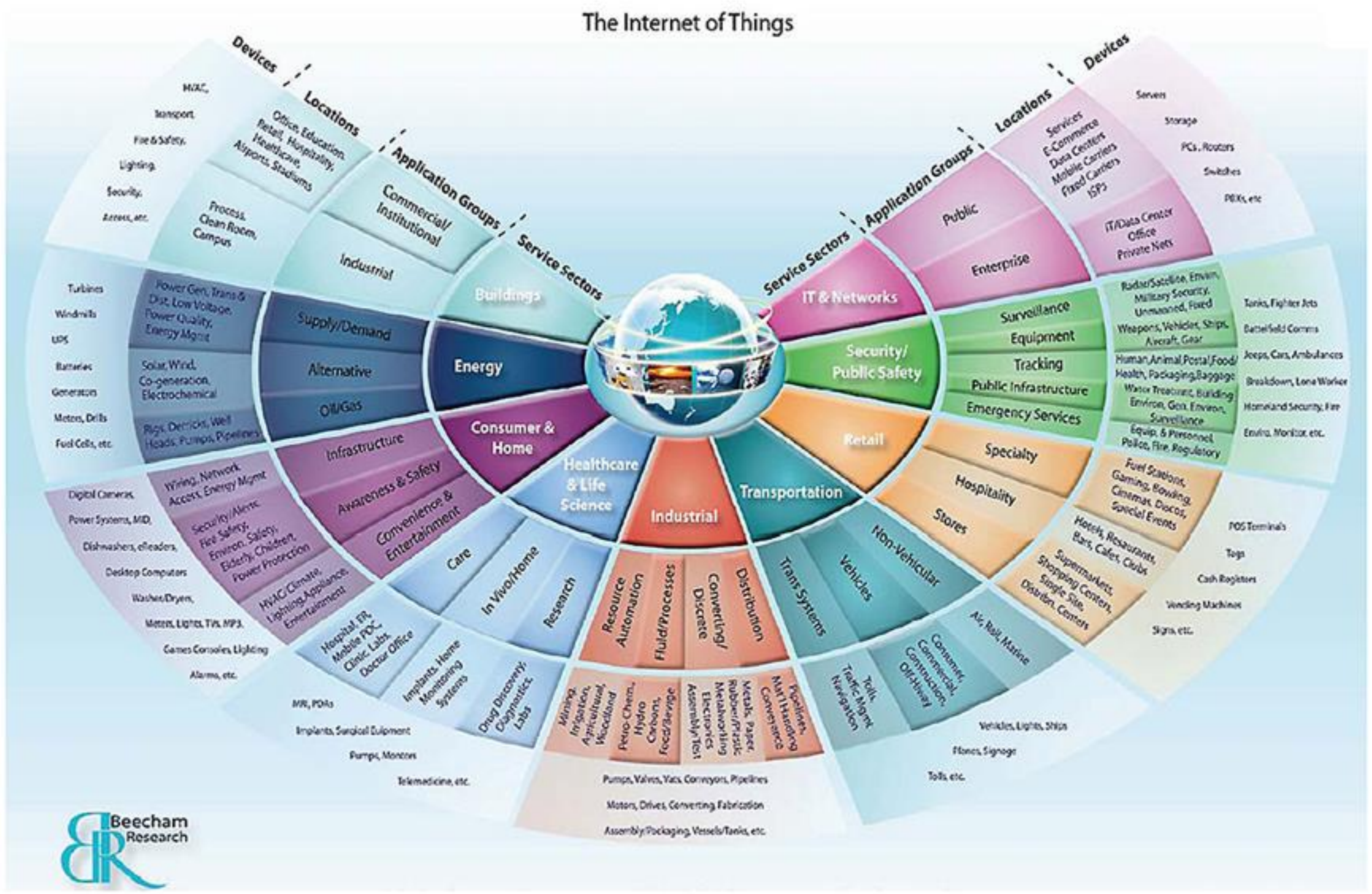
Source: QMEE, PC Mag.com, Go-Gulf.com, Business Insider, MailOnline.com, 4MAT, Intel (Jul 2013)

IMPACTING

EVERYONE, EVERYTHING, EVERYWHERE, EVERY DAY

Source: Cisco Virtual Networking Index (Feb 2014), Wikipedia, BBC Future (Jun 2013)

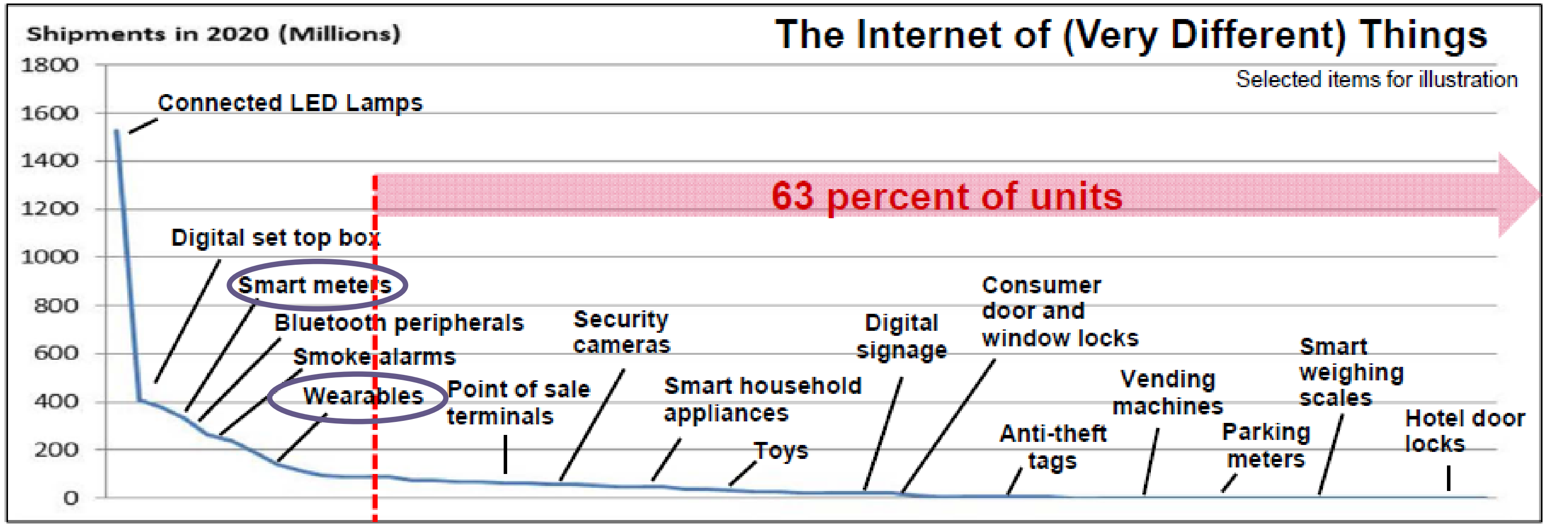
Internet of Things (IoT) – Fragmentation


The Internet of Things
Devices
Locations
Application Groups
Service Sectors
HVAC,
Transport,
Fire & Safety,
Lighting,
Security,
Access, etc.
Office, Education, Retail, Hospitality, Healthcare, Airports, Stadiums
Process, Clean Room, Campus
Commercial/ Institutional
Industrial
Buildings
Turbines
Windmills
UPS
Batteries
Generators
Meters, Drills
Fuel Cells, etc.
Power Gen, Trans & Dist, Low Voltage, Power Quality, Energy Mgmt
Solar, Wind, Co-generation, Electrochemical
Rigs, Derricks, Well Heads, Pumps, Pipelines
Supply/Demand
Alternative
Oil/Gas
Energy
Digital Cameras,
Power Systems, MID,
Dishwashers, eReaders,
Desktop Computers
Washer/Dryers,
Meters, Lights, TVs, MP3,
Games Consoles, Lighting
Alarms, etc.
Wiring, Network Access, Energy Mgmt
Security/Alerts, Fire Safety, Environ. Safety, Elderly, Children, Power Protection
HVAC Climate, Lighting, Appliance, Entertainment
Infrastructure
Awareness & Safety
Convenience & Entertainment
Consumer & Home
Hospital, ER, Mobile POC, Clinic, Labs, Doctor Office
Implants, Home Monitoring Systems
Drug Discovery, Diagnostics, Labs
Care
In Vivo/Home
Research
Healthcare & Life Science
MRI, PDAs
Implants, Surgical Equipment
Pumps, Monitors
Telemedicine, etc.
Resource Automation
Fluid/Processes
Converting/ Discrete
Distribution
Industrial
Mining, Irrigation, Agricultural, Woodland
Petro-Chem, Hydro Carbons, Food/Beverage
Metals, Paper, Rubber/Plastic, Metalworking, Electronics Assembly/Test
Pipelines, Mat'l Handling, Conveyance
Pumps, Valves, Vats, Conveyors, Pipelines
Motors, Drives, Converting, Fabrication
Assembly/Packaging, Vessels/Tanks, etc.
Transportation
Trans Systems
Vehicles
Non-Vehicular
Tolls, Traffic Mgmt, Navigation
Consumer, Commercial, Construction, Off-Hwy
Air, Rail, Marine
Vehicles, Lights, Ships
Planes, Signage
Tolls, etc.
Retail
Specialty
Hospitality
Stores
Fuel Stations, Gaming, Bowling, Cinemas, Discos, Special Events
Hotels, Restaurants, Bars, Cafes, Clubs
Supermarkets, Shopping Centers, Single Site, Distribn. Centers
POS Terminals
Tags
Cash Registers
Vending Machines
Signs, etc.
Security/ Public Safety
Surveillance
Equipment
Tracking
Public Infrastructure
Emergency Services
Radar/Satellite, Environ, Military Security, Unmanned, Fixed
Weapons, Vehicles, Ships, Aircraft, Gear
Human, Animal, Postal, Food/ Health, Packaging, Baggage
Water Treatment, Building Environ, Gen. Environ, Surveillance
Equip. & Personnel, Police, Fire, Regulatory
Tanks, Fighter Jets
Battlefield Comms
Jeeps, Cars, Ambulances
Breakdown, Lone Worker
Homeland Security, Fire
Enviro. Monitor, etc.
IT & Networks
Public
Enterprise
Services, E-Commerce, Data Centers, Mobile Carriers, Fixed Carriers, ISPs
IT/Data Center, Office, Private Nets
Servers
Storage
PCs, Routers
Switches
PBXs, etc
Beecham Research

Expect Many Types of Things; Highly Fragmented Market

By 2017, 50% of Internet of Things solutions will originate in startups less than three years old.

- Expect 10 billion shipments in 2020
- Many smart versions of existing product markets
- Key challenge: where to focus?



* Preliminary, September 2013

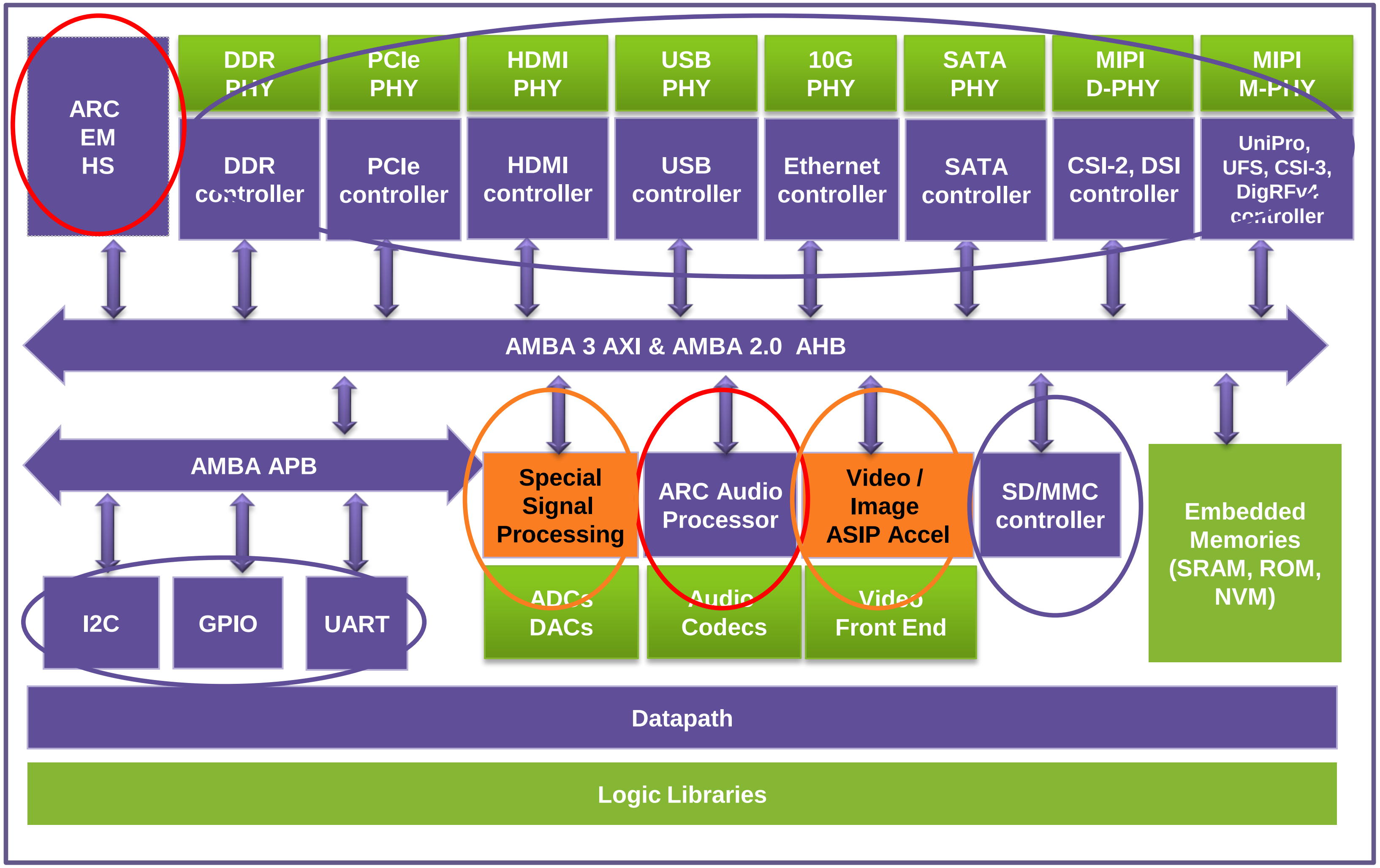
6

Gartner

SYNOPSYS® Accelerating Innovation

Complementary All-SoC Offering

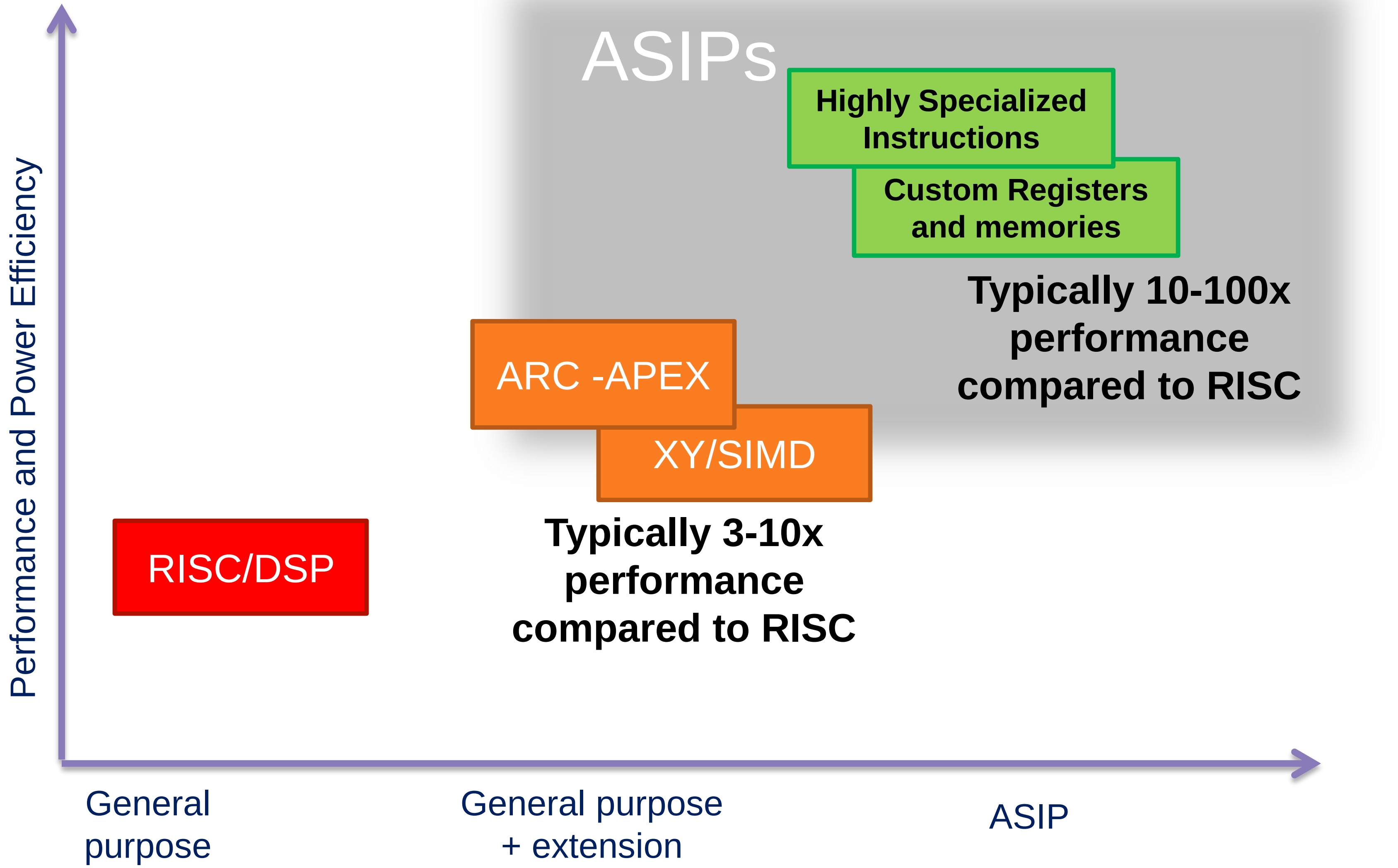
Processor IP ***Physical/Digital IP*** ***ASIPs***



Digital IP | Physical IP | Customer Design

Application Specific Processors (ASIP)

Optimizing for Specific Applications


ASIPs
Highly Specialized Instructions
Custom Registers and memories
Typically 10-100x performance compared to RISC
ARC -APEX
XY/SIMD
Typically 3-10x performance compared to RISC
RISC/DSP
Performance and Power Efficiency
General purpose
General purpose + extension
ASIP

“No MPSoC Design Without Tools”

- **Tools at IP level (ASIP cores)**
  - Architectural exploration
  - SDK generation: C compiler, ISS, debugger…
  - RTL generation

  → ***IP Designer***

- **Tools at IP subsystem level (multi-core)** ← **This presentation**
  - Code parallelisation
  - Communication and synchronization
  - Multi-core platform generation

  → ***MP Designer***

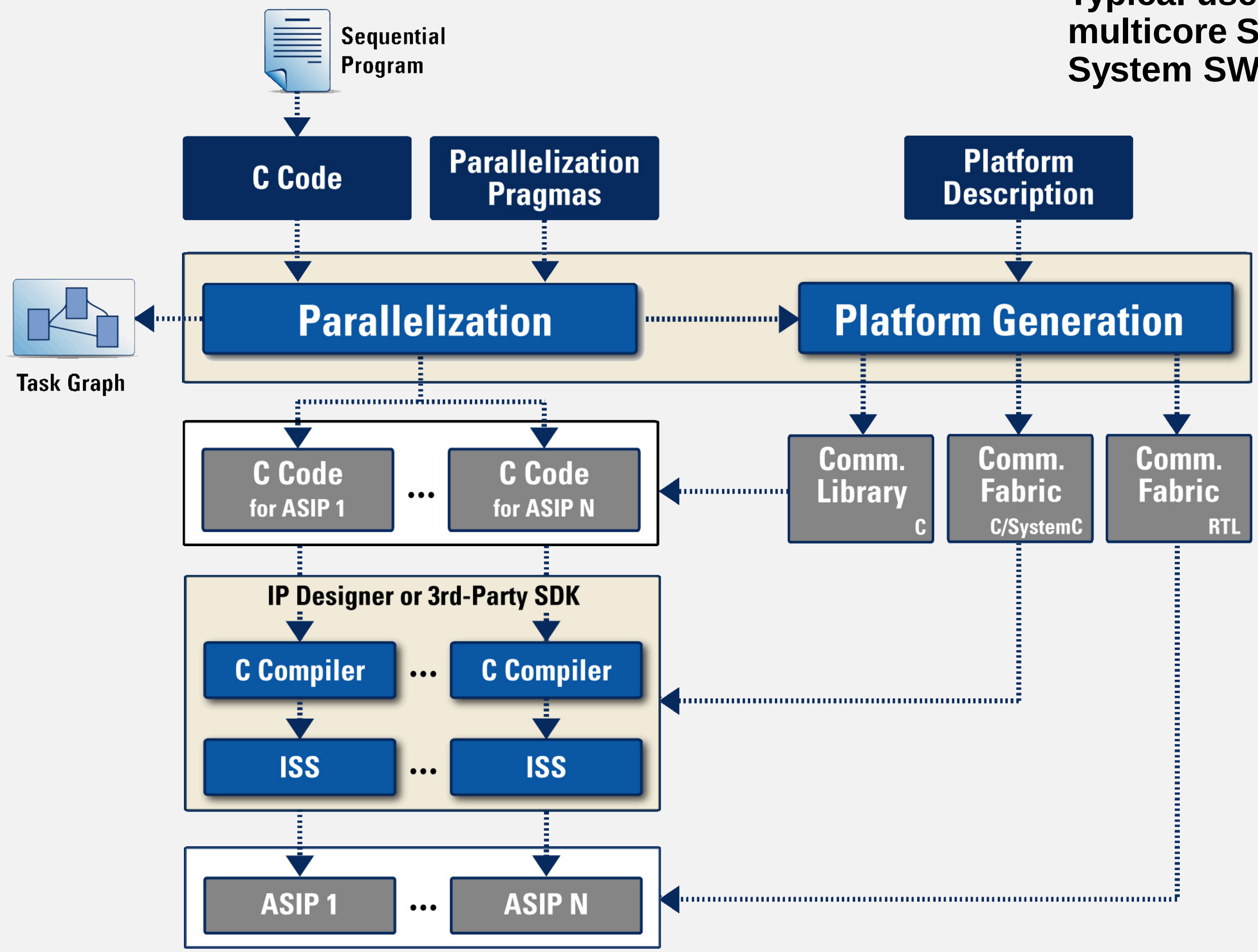
When is MP Designer Used?

- **When the application is coded in sequential C code (for single-core execution), but parallelism must be introduced to improve performance or power**
  - Targeting “multi-core” (≤ 10 cores) rather than “many-core” (e.g. ~100 cores)
    - Many-core is better served by languages with parallelism support, e.g. OpenCL
  - Application must allow for static parallelization (i.e. decided at design time)
  - Application must benefit from task-level parallelism (e.g. pipelined execution of tasks) and/or data-level parallelism* (parallelize loop iterations)
- **Exploration is needed to achieve efficient load-balancing and low communication cost**
- **Option for specializing the individual core architectures**
  - Bring IP Designer (ASIP design tool) in the loop
  - May result in heterogeneous multi-core architecture
- **Option for generating a custom communication fabric**
  - Using point-to-point connections

* Next release

MP Designer Tool-Suite

Typical users:
multicore SoC design teams
System SW design teams



User-Guided Parallelization

```
int main(int argc,char *argv[]){
  init_all();
  parsection: {
    jpg_open: {
      jpg_fopen(JPG_filename);
      writeword(0xFFD8); //SOI
      write_APP0info();
    }
    main_encoder(&in_img);
    jpg_close: {
      writeword(0xFFD9); //EOI
      jpg_fclose();
    }
  }
  free(in_img.RGB_buffer);
  return 0;
}

void main_encoder(struct image* img) {
  vlc_init: { DCY=0;DCCb=0;DCCr=0; }
  for (ypos=0..height) {
    for (xpos=0....width) {
      for (blk=0..5) {
        SBYTE DU[64];
        loading:
          load_data_unit_from_RGB_buffer(img,
            xpos, ypos, blk, DU);
        process_DU(DU,blk);
      }
    }
  }
  vlc_fini: {    // Bit-alignment of EOI marker
    if (bytepos>=0) {
        writebits((1<<(bytepos+1))-1, bytepos+1);
    }
  }
}
...
```

Example: high-res JPEG encoding on 3-DLX architecture

← **C code**
labels added for parallelization

↓ **Parallelization pragmas**
pragmas referring to C labels

```
processor P0 type dlx
processor P1 type dlx
processor P2 type dlx

parallel ParRegion lbl main::parsection

  task LOAD
    target P0
    include lbl main_encoder::loading

  task DCT
    target P1
    include lbl process_DU::fdct_main

  task VLC
    target P2
    include lbl main::jpg_open
    include lbl main_encoder::vlc_init
    include lbl process_DU::vlc_main
    include lbl main_encoder::vlc_fini
    include lbl main::jpg_close
```

User-Guided Parallelization

- **Exploring parallelization choices is easy and fast**
  - Always work on sequential C code
  - Add source-code labels and parallelization pragmas
  - For each parallelization choice, MP Designer:
    - Checks all dataflow dependencies
    - Adds all required communication and synchronisation code, using a FIFO communication model
    - Provides feedback about performance and load balancing, memory and communication cost, data dependencies

FIFO Communication

Sequential C code

```
void foo() {
  int A[100];
producer:
  for (...) {
    A[i] = ...;
  }
consumer:
  for (...) {
    ... = A[i];
  }
}
```

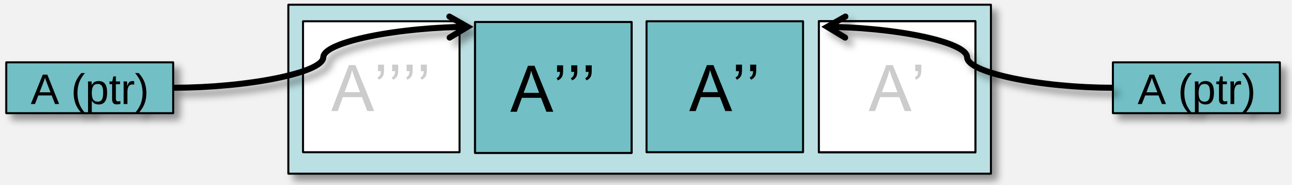
MP Designer

Parallelized C code producer

```
// Producer side
DEFINE_SRC_FIFO(A,int,100)
void foo() {
  int* A;
producer:
  A = FIFO_A_acq_put();
  for (...) {
    A[i] = ...;
  }
  FIFO_A_rel_put();
}
```

Parallelized C code consumer

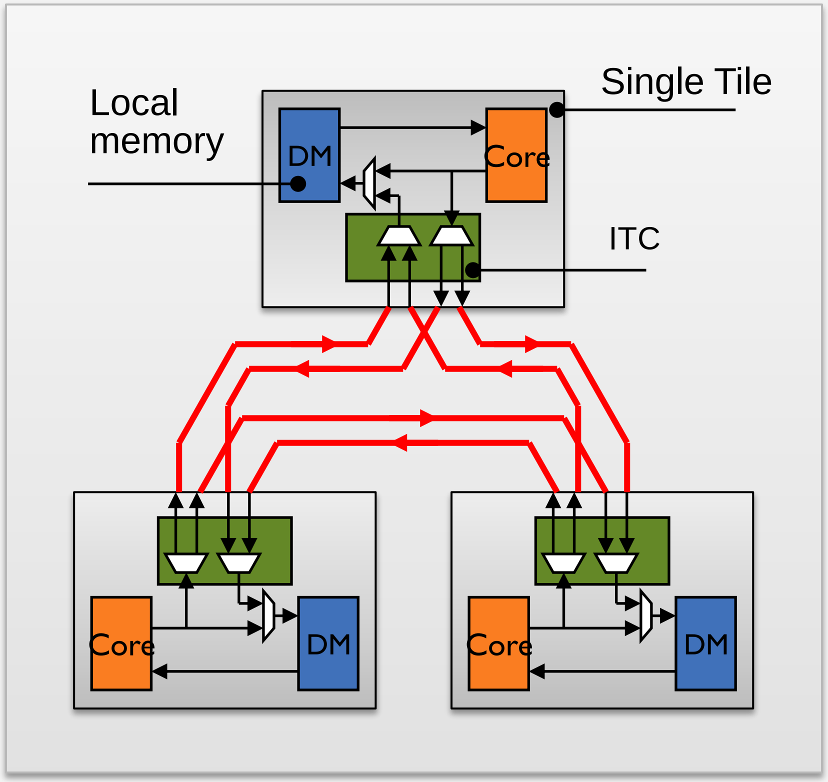
```
// Consumer side
DEFINE_DST_FIFO(A,int,100)
void foo() {
  int* A;
consumer:
  A = FIFO_A_acq_get();
  for (...) {
    ... = A[i];
  }
  FIFO_A_rel_get();
}
```



- **Acquire/release interface** enables use of other processor's data memory for storage of arrays (avoiding local copies)
- Synchronization implemented by polling on FIFO queue's status (empty/full)
- Address translation for communicated pointers

Communication Fabric

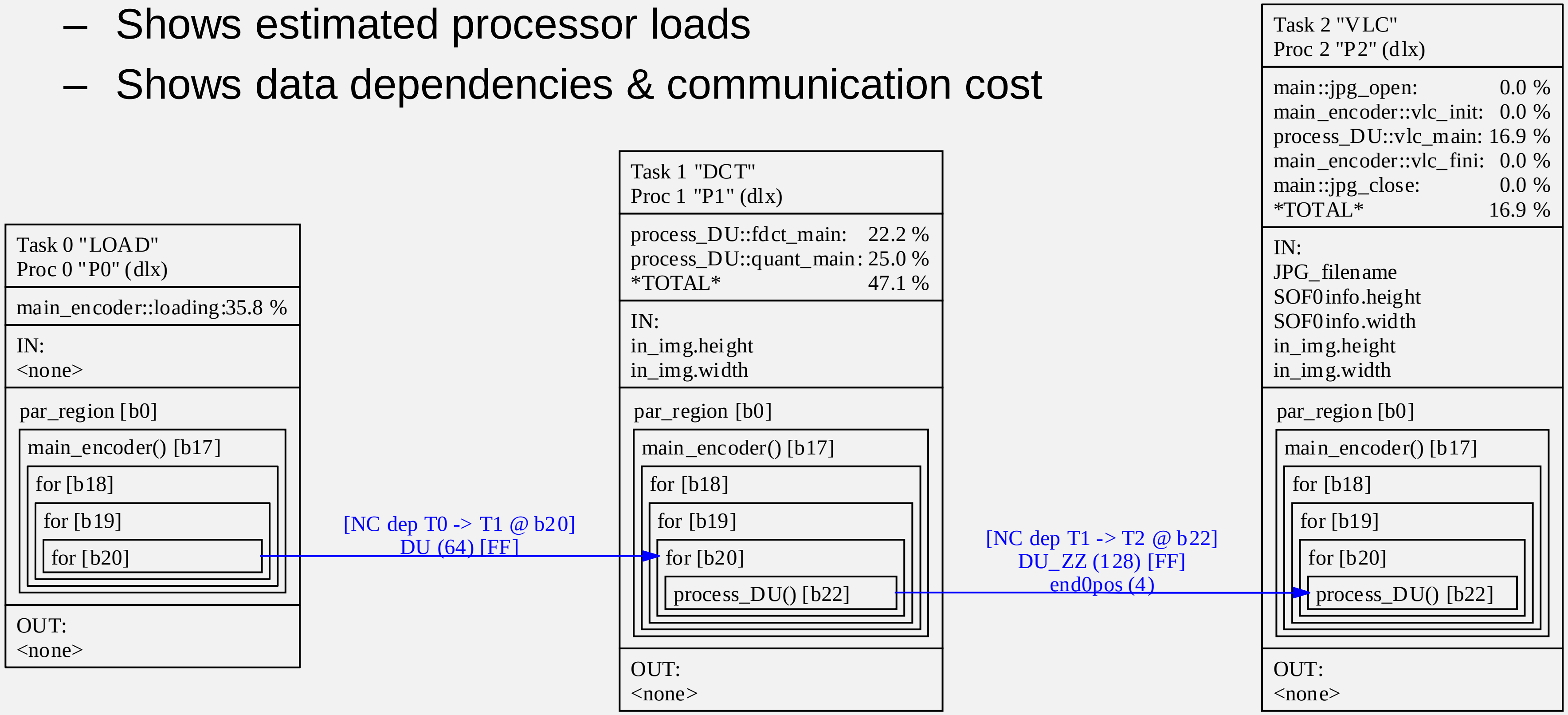
- **Current**
  - Point-to-point links between communicating processors
  - Communication FIFOs mapped into destination processor's local data-memory
  - Write conflicts resolved through local buffering
  - Address decoding logic
  - User constraints
- **Future**
  - Shared memory, shared bus
- **RTL & simulation model generated**



Exploration

Task Graph

- **MP Designer generates a task graph for each parallelization alternative**
  - Shows estimated processor loads
  - Shows data dependencies & communication cost

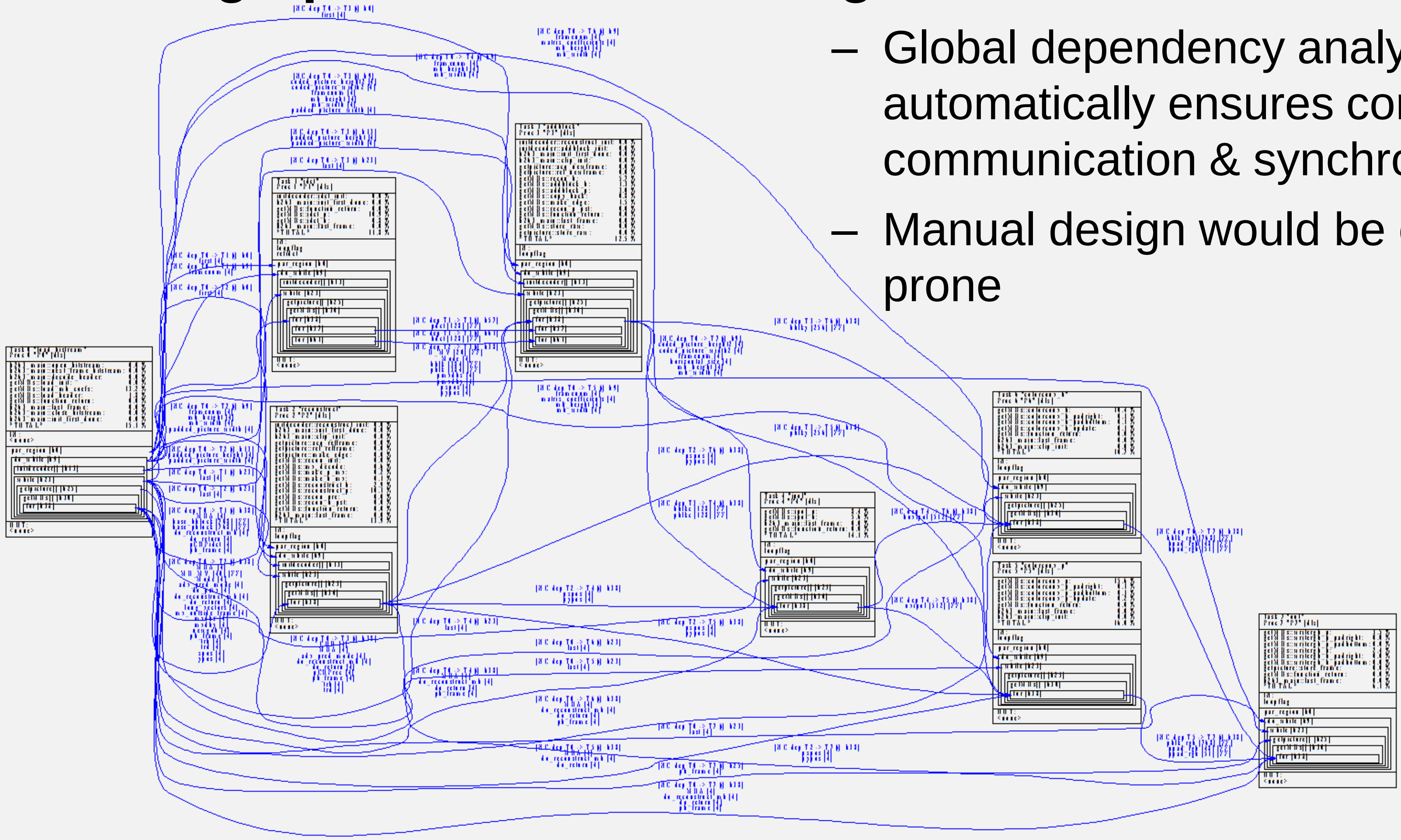


High-res JPEG encoding on 3-DLX architecture

Exploration

Task Graph

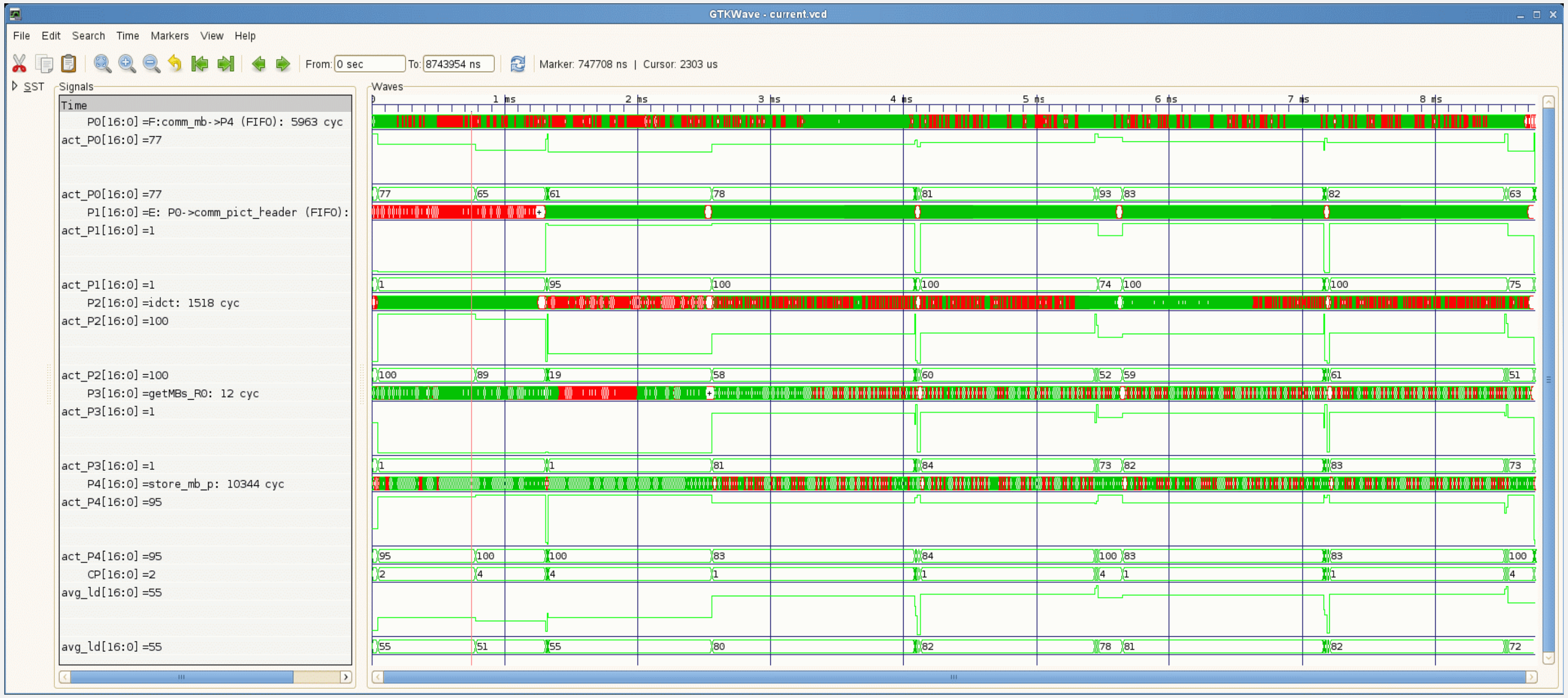
- **Task graph for H263 encoding on 8 cores**
  - Global dependency analysis automatically ensures correct communication & synchronisation
  - Manual design would be error-prone



Exploration

Activity Diagram

- **MP Designer generates an activity diagram for each parallelization alternative**
  - Dynamic view of core utilization


GTKWave - current.vcd
File Edit Search Time Markers View Help
From: 0 sec
To: 8743954 ns
Marker: 747708 ns | Cursor: 2303 us
SST
Signals
Waves

```
Time
    P0[16:0] =F:comm_mb->P4 (FIFO): 5963 cyc
act_P0[16:0] =77
act_P0[16:0] =77
    P1[16:0] =E: P0->comm_pict_header (FIFO):
act_P1[16:0] =1
act_P1[16:0] =1
    P2[16:0] =idct: 1518 cyc
act_P2[16:0] =100
act_P2[16:0] =100
    P3[16:0] =getMBs_R0: 12 cyc
act_P3[16:0] =1
act_P3[16:0] =1
    P4[16:0] =store_mb_p: 10344 cyc
act_P4[16:0] =95
act_P4[16:0] =95
    CP[16:0] =2
avg_ld[16:0] =55
avg_ld[16:0] =55
```

H263 encoding on 5-DLX architecture

Exploration

- **JPEG encoding on multi-DLX architecture**

Algorithm	# Cores	Parallelization	Mcycles* seq	Mcycles* par	Speed up	Load (%) P0	Load (%) P1	Load (%) P2	Load (%) P3	Load (%) P4	Effi-ciency (%)
Original	1		7.1	-	1	100					100
Original	2	ld+dct+q | vlc	7.1	4.1	1.7	100	76				86
Original	3	ld | dct+q | vlc	7.1	3.4	2.1	64	60	91			69
Original	4	ld | dct | q | vlc	7.1	3.4	2.1	77	40	24	91		52
Optimised	2	ld+dct | q+vlc	4.1	2.4	1.5	100	72				85
Optimised	3	ld | dct+q | vlc	4.1	2.0	2.0	74	100	40			68
Optimised	3	ld | dct | q+vlc	4.1	1.7	2.4	86	56	100			78
Optimised	4	ld | dct | q | vlc	4.1	1.5	2.8	100	65	75	54		69
Split quant	3	ld | dct+q0 | q1+vlc	4.3	1.6	2.6	92	100	79			87
Dual load	5	ld0 | ld1 | dct | q | vlc	4.1	1.1	3.7	71	61	95	99	71	74

* Cycles for 256x160-pixel image

  – Entire exploration in only days of time

Summary

- **IoT will drive many different MultiCore SoCs**
- **No (efficient) multicore SoC design without tools**
  - Design and programming of individual ASIP cores
  - Multicore parallelisation and platform generation
- **MP Designer tool-suite**
  - Parallelisation from sequential C code
  - Exploration of functional parallelism
  - Static global dependency analysis
  - Efficient FIFO communication model (acquire/release interface)
  - All communication and synchronization code added automatically
  - Feedback about performance and load balancing, memory and communication cost, data dependencies

Thank You

SYNOPSYS®

Accelerating Innovation