

Research Network for System Level Design of Embedded Systems: Dynamic Memory Allocation Design Flow Case Study

S. Mamagkakis, D. Atienza, F. Catthoor,
D. Soudris and J. Mendias



Research Network for System Level Design of Embedded Systems



Motivation:

IMEC as a bridge between industry and academia

Industry



IMEC vzw.



Universities



- Embedded system design solutions that are relevant for the industry
- Critical mass of researchers is needed to solve bigger, longer term problems
- Common problem definition and priorities are necessary

DACMA Marie-Curie star shaped network (PhD candidate collaboration in Europe)

- PhD candidates visit IMEC for a minimum of 3 months
- Their visits are fully funded by Marie-Curie scholarships
- They can have multiple visits (no more than half of their PhD duration)
- PhDs are trained by IMEC, collaborate with researchers and among themselves
- Their PhD focuses on a specific aspect of a bigger problem in Embedded System design
- Collaborative effort
- Each PhD contributes according to his specialty



Research network has been a success story

- Started in 1995 by F. Catthoor
- EC funding was later provided by DATMA Marie-Curie project
- EC Funding is extended with the DACMA Marie-Curie project (follow-up)
- More than 50 PhDs were trained and collaborated until now
- Each year ~10 PhDs visit IMEC for 3-4 months (40 person months per year)
- Absorbance rate 90%-100%

Training and collaboration of sandwich PhDs

- **Training**

- Presentations from external prominent researchers every Friday (from industry and academia)
- Presentations from internal IMEC-researchers every Tuesday
- 3-day seminars about new design technologies/standards
- Other seminars (e.g., public speaking, technical writing, etc.)
- Daily supervision/training about the integrated design flow (terminology issues, shared problem perspective, etc.)

- **Collaboration**

- With IMEC-researchers (~200 researchers for DESICS division)
- With international residents (mainly industrial partners, see www.imec.be)
- With other sandwich PhDs (from 15 Universities and 3 research centers)

Big gains for sandwich PhDs

- Training opportunities
- **Benefit from a big collection of specialists in every field of Embedded System design**
- Gain further qualifications through seminars
- **Extend their skills by collaboration with other PhD candidates with similar topics**
- Common Publications (e.g., more than 30 publications for 2005)
- International collaborations
- Networking opportunities
- **High industrial relevance**

Viral effect (dissemination continues in the home country of each student in their group)

What do the students think?

Overall success: very good

Research assessment:

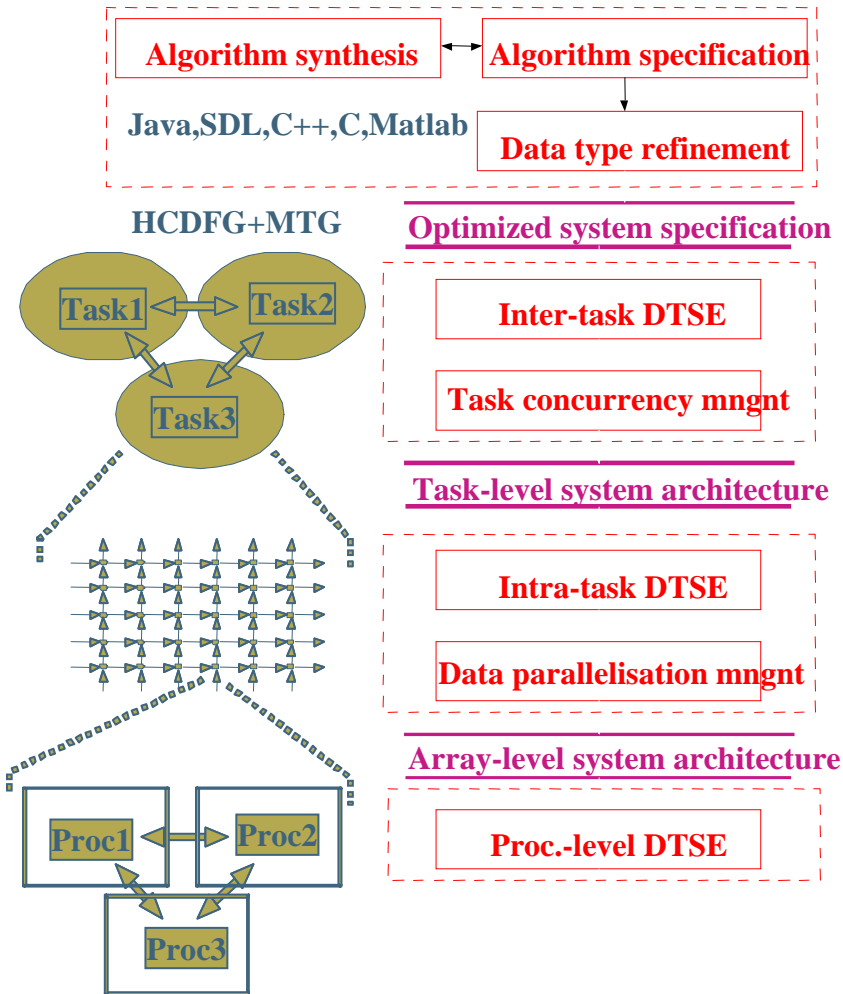
- participation in meetings: good
- discussion of results: very good
- co-operation within team: very good
- co-operation with other host members: good
- originality: good
- capacity to develop skills: good
- productivity: very good
- communication skills: good
- group leader skills: good
- training/teaching: good

(average score of 10 students from the 2005 DACMA report)

Dynamic Memory Allocation Design Flow Case Study

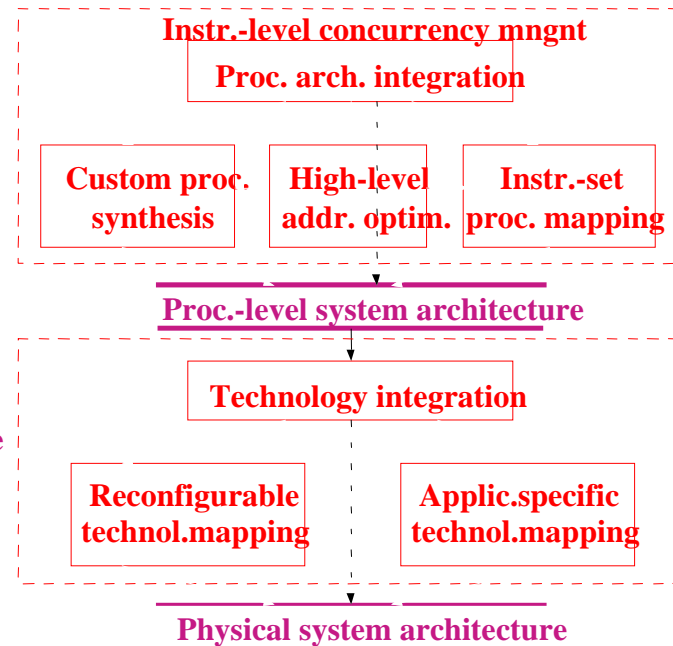


Unified meta-flow

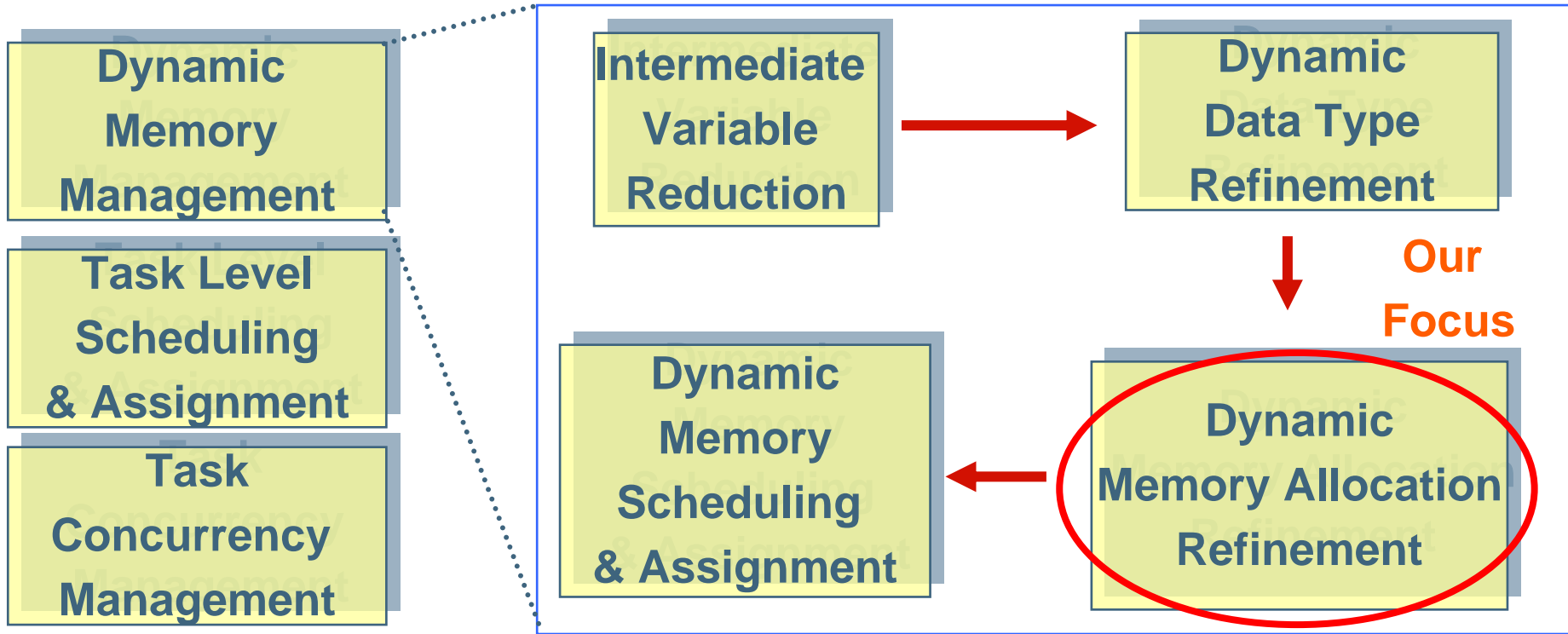


Reference:

F. Catthoor et al., *“Unified meta-flow summary for low-power data-dominated applications”*, Kluwer Publications, 2000.

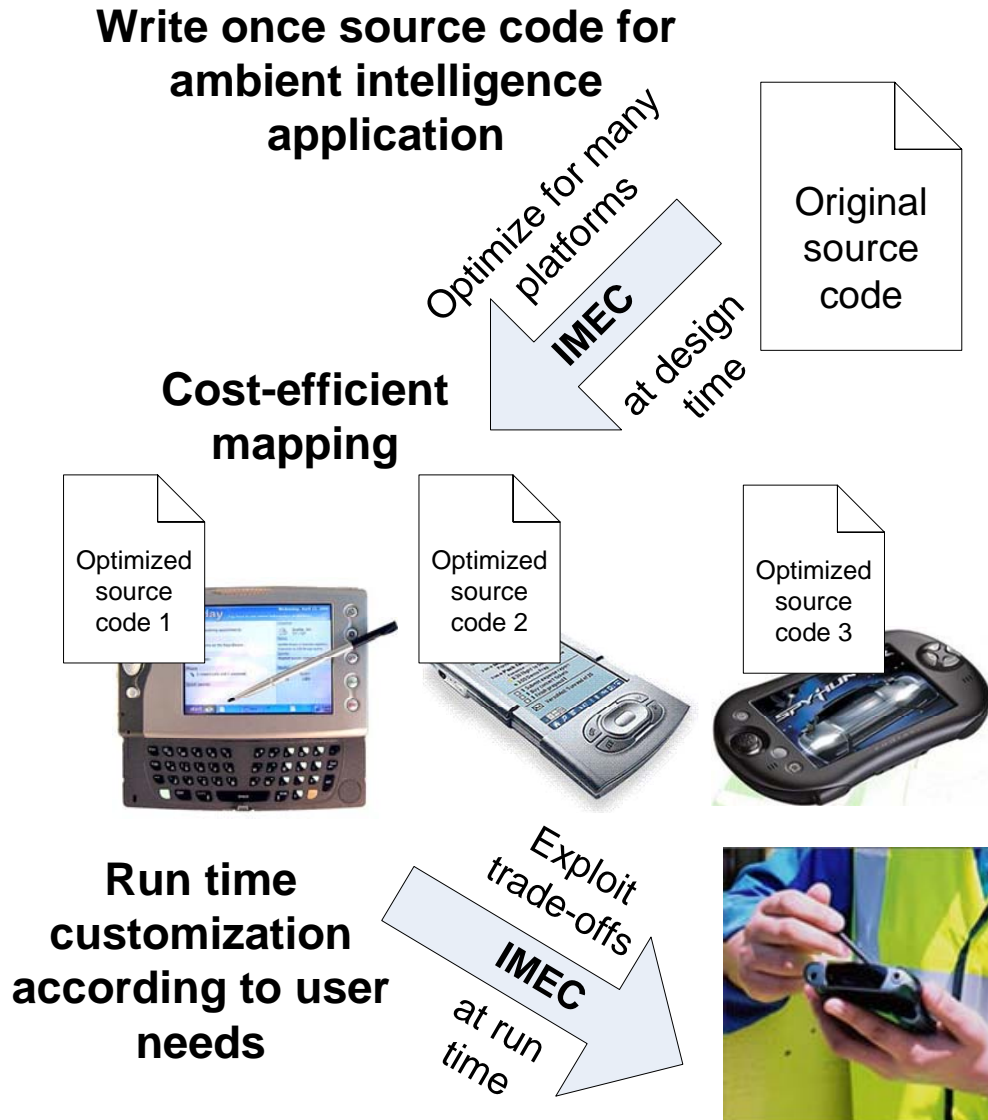


We focus on a specific part of the global design flow



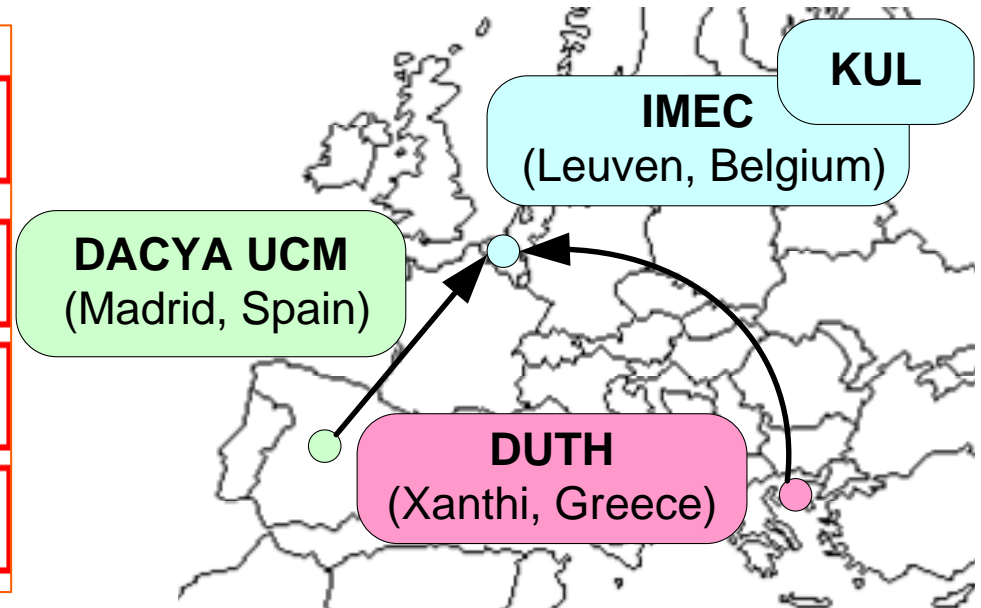
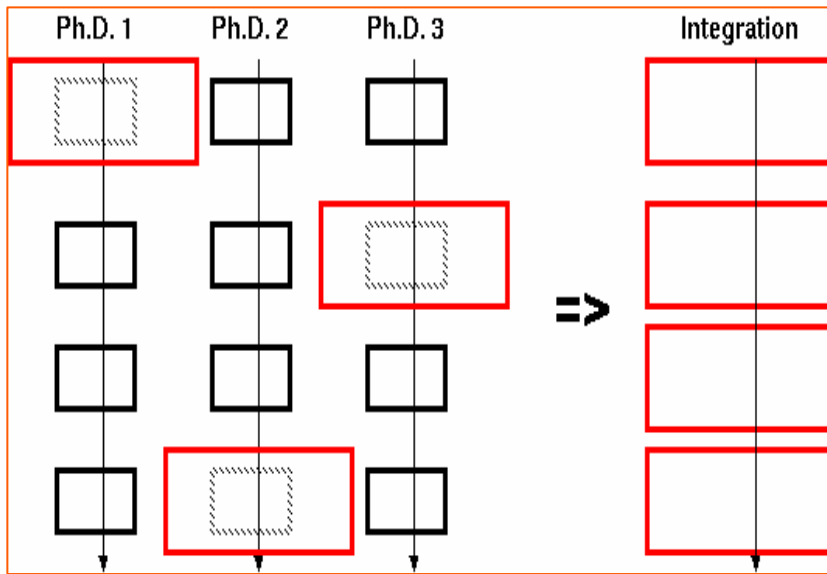
The focus of the collaboration was **Dynamic Memory Allocation Refinement** (malloc/free, garbage collection, new/delete)

Our Motivation



- New **dynamic applications** of wireless multimedia terminals have increased needs of data transfer and storage
- Dynamic **memory management** of an embedded system is a **critical** part of the final design
- A successful design and implementation of a **customized** dynamic memory management **solution** guarantees lower energy consumption, small size and increased speed

Dynamic Memory (DM) Allocation Refinement Collaboration

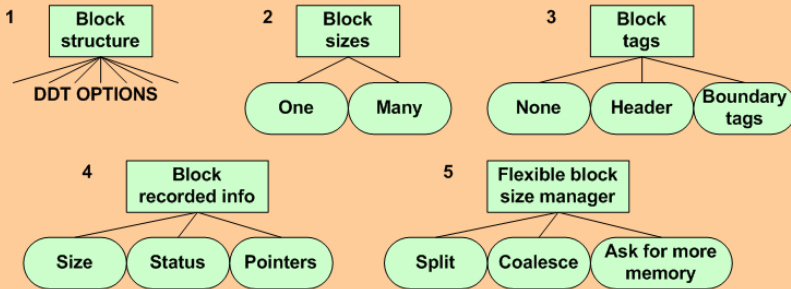


3 PhD candidates were involved:

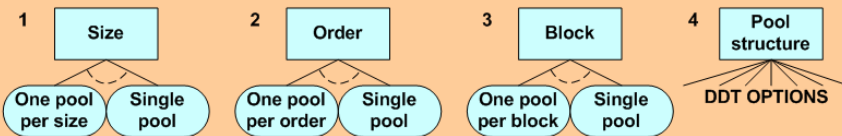
- S. Mamagkakis (Greece)
- D. Atienza (Spain)
- M. Leeman (Belgium)

Huge design space of available solutions

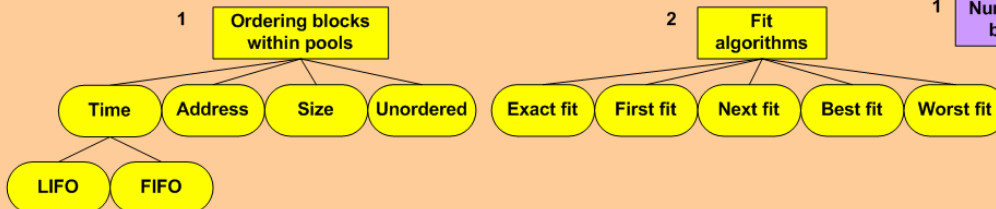
A. Creating block structures



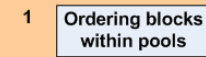
B. Pool division based on criterion



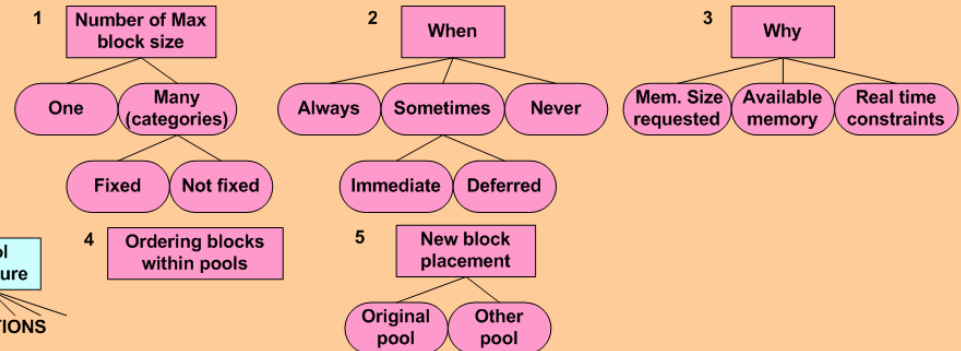
C. Allocating Blocks



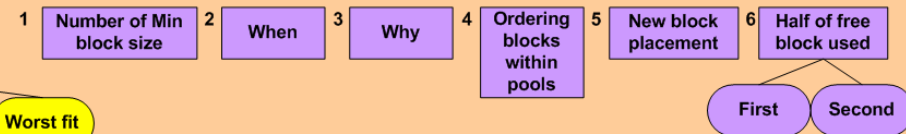
D. De-allocating Blocks



E. Coalescing Blocks

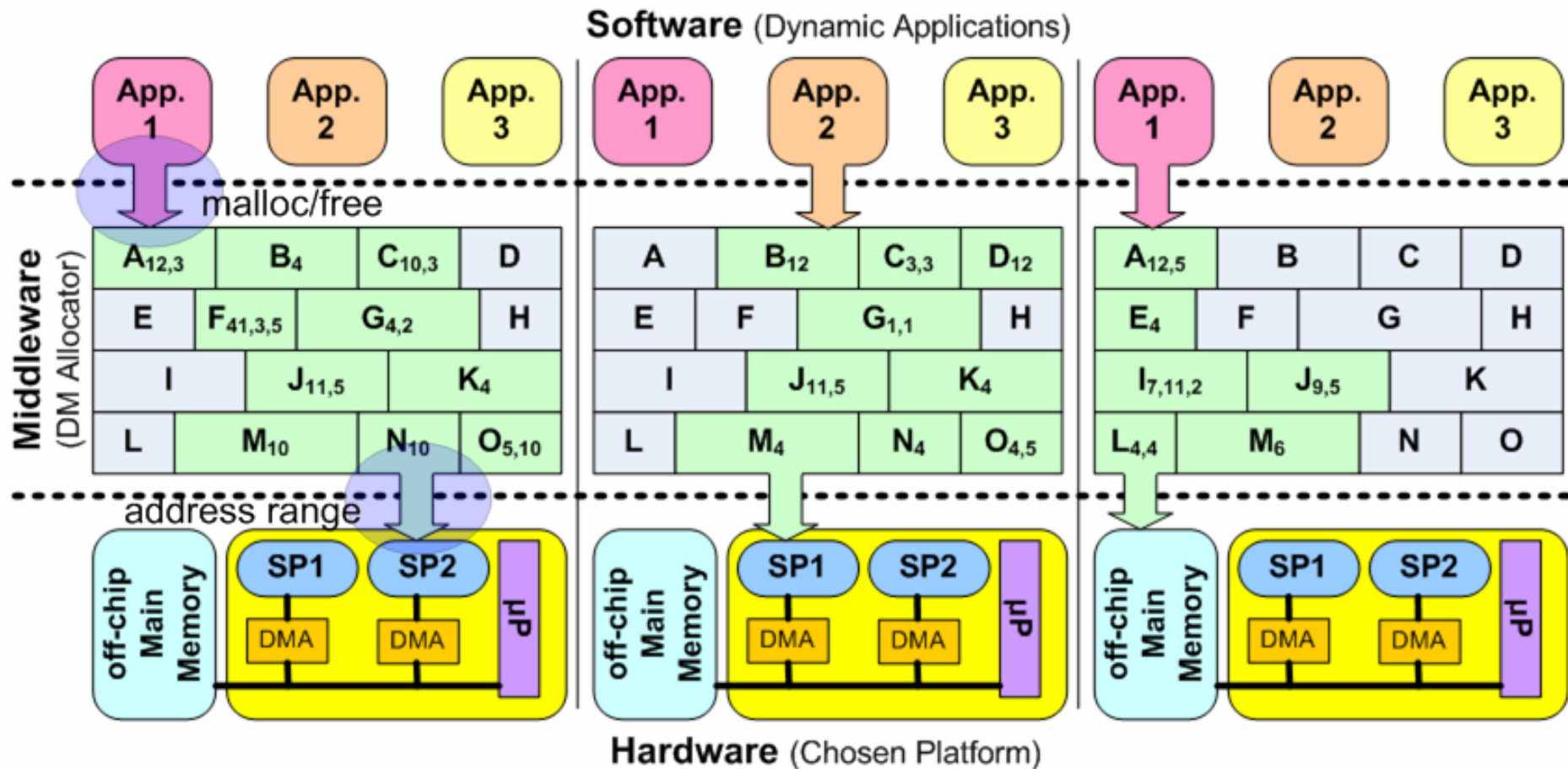


F. Splitting Blocks



With the combination of proposed solutions, we are able to create millions of ultra-customized DM allocators

Super-customized DM allocators (according to SW application and HW platform)



Customization of DM allocators:

- According to applications (software)
- According to memory hierarchy (hardware)

Our Tool for automatic exploration of DM allocator solution

Prototype Custom Dynamic Memory Allocation Tool

Automatic Custom Dynamic Memory Allocation Parameter Exploration:

- Block Coalescing support** *Max. coalesced block size (KB):*
How much to coalesce (%):
- Block Splitting support** *Min. produced block size (KB):*
How much to split (%):
- Ordering of the blocks within pool: LIFO FIFO Address ordered
 - Begin at high address
 - Begin at low address
- Fit algorithm used to choose inside pool: Exact Fit First Fit Next Fit
- Dynamic Data Type Connection of blocks within pool:
 - Dynamic Array (DA) Doubly linked list (DLL) Single
 - DA of DLLs DLL of DAs
 - DA of SLLs DLL of SLLs
- Freelists (input the size of the blocks to be stored in KBs):
 - Freelist 1: 0.066, 1.514, 1.515, 1.51 Freelist 2
 - Freelist 5 Freelist 6
 - Freelist 9 Freelist 10
 - Freelist 13 Freelist 14
- Proposed custom DM manager Explorations for this application:
 - Propose a custom DM manager exploration for low mem. footprint:
 - Propose a custom DM manager exploration for low mem. accesses:

Simulation

Execution time (msec) vs Energy Consumption (nJ) (10⁻³)

Energy Consumption (nJ) (10⁻³) vs Mem. footprint (Bytes) (10⁻⁴)

Mem. footprint (Bytes) (10⁻⁴) vs # Accesses (10⁻³)

Accesses (10⁻³) vs Execution time (msec) (10⁻⁰)

Producing the definition of the customized Dynamic Memory Manager...
 Simulating the application without instrumented code to get execution time results...
 Compiling the application with the Customized Dynamic Memory Manager 7...

Prototype Custom Dynamic Memory Allocation Tool:

Insert File with the application's main() function

je/smagka/Amdrel-final-review-meeting-demo/demo_date06/easyport/eas

Start initial run-time profiling

Finding malloc() and free() within the application and putting instrumentation...
 Doing initial run-time profiling...
 Processing initial profiling results...

Initial run-time profiling results

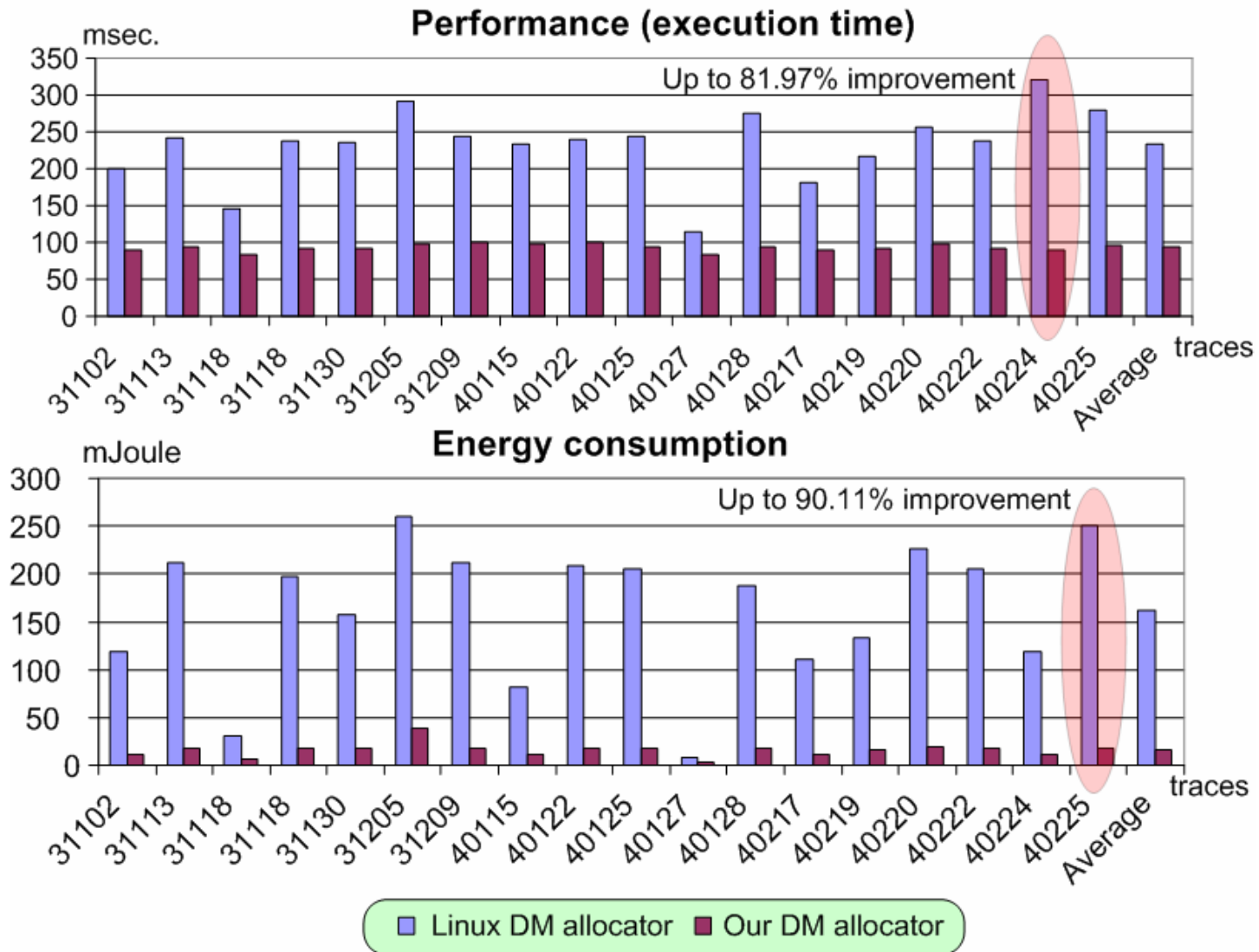
```

#num-accesses = 43539
#num-mallocs = 972
#num-frees = 228
#max-blocks-active = 744
#scopes = 1
BlockAccesses =
-----
pool-id  block-size  #mallocs  #frees  #data-accs  malloc-effort
-----
1        66          245       54      0            7771
1        72          6         6       0            198
1        74          18        18      0            582
1        79          6         6       0            198
1        82          12        12      0            390
        
```

Automatic Dynamic Memory Allocation Parameter Exploration

Edit memory hierarchy settings

Very promising results (802.11b example)



Dissemination:

of Publications of the partners the last 4 years

- 3 internal technical reports
- 1 Book Chapter
- 6 Journals
 - 1 IEEE
 - 1 ACM
 - 1 IEE
 - 3 Elsevier
- 19 Conference Papers (eg., 4 in DATE conf., 1 in DAC conf. - candidate for best paper award)
- Currently we are preparing a book about 'Dynamic Memory Management'

Conclusions

- Collaboration can solve bigger, longer term problems in Embedded System Design
- It requires an initial investment in training
- There is a small management/integration overhead
- The final results are very promising and justify the investment

IMEC's 'sandwich PhD' model can be re-used by other institutes around the world!

aspire invent achieve

