# Labex UCN@Sophia

http://ucnlab.eu

# High level design of low power communicating devices

Emilien Kofman under the supervision of Robert de Simone (INRIA) and François Verdier (LEAT)

# Outline

- Me
- My thesis
  - Modeling embedded systems
  - Modeling software and hardware
  - Results
  - Current and future work

# Me

- **Jan 2013**: Engineer's degree from Telecom SudParis (ex INT)
  - **Sept 2012**: Last year in EURECOM

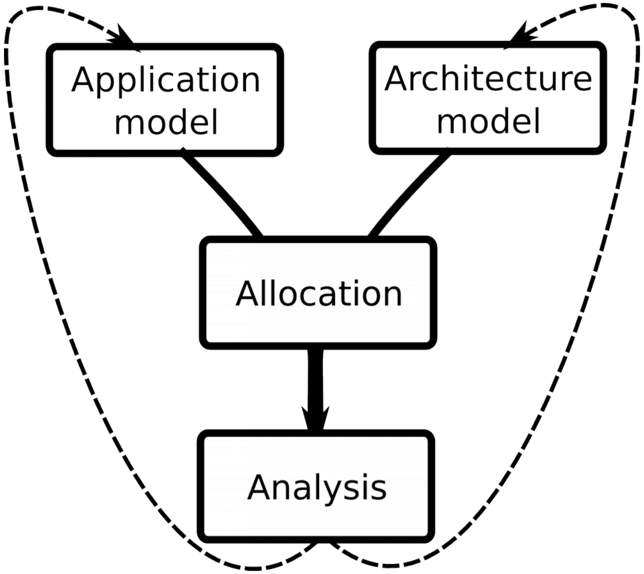
    Embedded and real time systems

    Intership in Qualisteo (embedded system signal processing)
- ~~**Expected 2013**: Thesis at Texas Instrument (Villeneuve Loubet), project HOPE~~. Aborted
- **2013**: CDD in Qualisteo, then in INRIA
- **2014**: Labex thesis + DCCE in Polytech'Nice

  Supervised by Robert de Simone (INRIA AOSTE UMR I3S) and Francois Verdier (LEAT). Concerns close to project HOPE.

# Modeling embedded systems

- Modeling hardware components, software components and constraints (HOPE project)
  - For real-time: deadline constraint of a reactive system
  - Low power consumption
  - Physical / temperature constraints

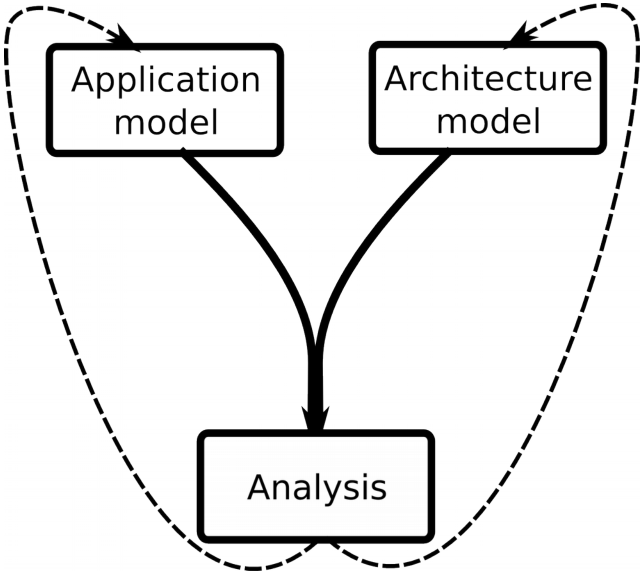


*Y-chart / AAA*

- Predict early
- Heuristics are not flexible, I need a framework in which I can change the objectives and/or models (eg add/remove a power model).

# Modeling embedded systems

- Modeling hardware components, software components and constraints (HOPE project)
  - For real-time: deadline constraint of a reactive system
  - Low power consumption
  - Physical / temperature constraints



*Y-chart / AAA*

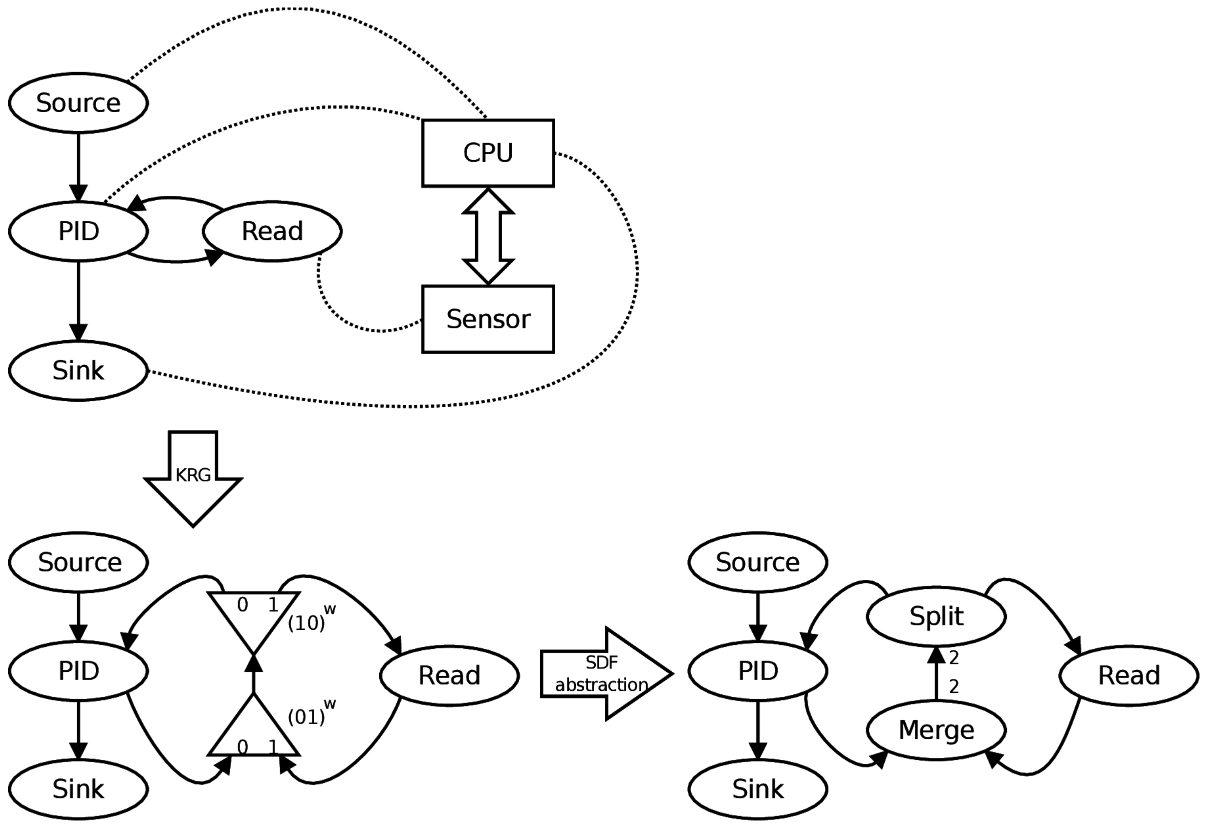
- Predict early
- Heuristics are not flexible, I need a framework in which I can change the objectives and/or models (eg add/remove a power model).

# Modeling embedded systems

- Previous work
  - Application Architecture Adequacy through an FFT case study (JRWRTC2013)
  - Modeling and analyzing dataflow applications on NoC based (TECS2014)
  - Efficient FFT mapping on GPU for radar processing application: modeling and implementation (not published)

# TECS2014

Idea: Use a model of computation to abstract both the application and the machine. Then prove safety properties on this system.



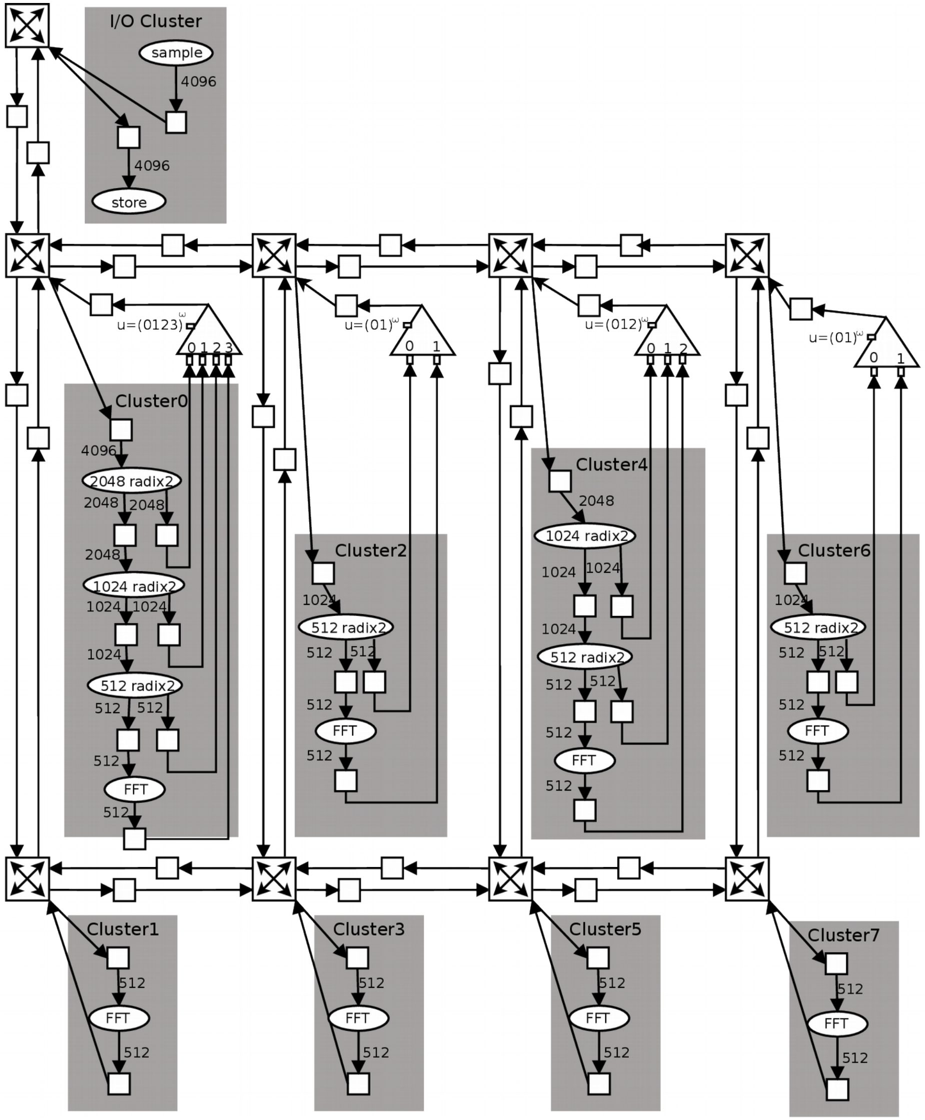
Modeling the interconnect is mandatory for predictability.

Embedded systems
→ not a shared machine
→ ability to control the interconnect.

(valid for some HPCs?)

Ex: static control of messages over a Bus.

# TECS2014



Example of modeling a 1D FFT over a mesh network architecture.

- Gives a partial order of software and hardware events.
- Ensures **safety properties** (deadlock/livelock detection)

but no optimality properties (e.g. here clustering is decided arbitrarily).

There is a notion of (partial) ordering in the application model (it's a DAG) but no notion of time (either physical or logical).

# State of the art offline scheduling

a.k.a. Operational Research ie “off line optimization of a scheduling and/or clustering problem”, may be homogeneous/heterogeneous preemptive/non-preemptive …

- With heuristics: HEFT, CPOP, … Some problems have near-optimal and polynomial time heuristics (for instance Bin-packing,). It is hard to design a heuristic for each model+objective.
- With exhaustive search: Scales poorly. CP / ILP / SAT is hopefully smart enough to solve small problems.

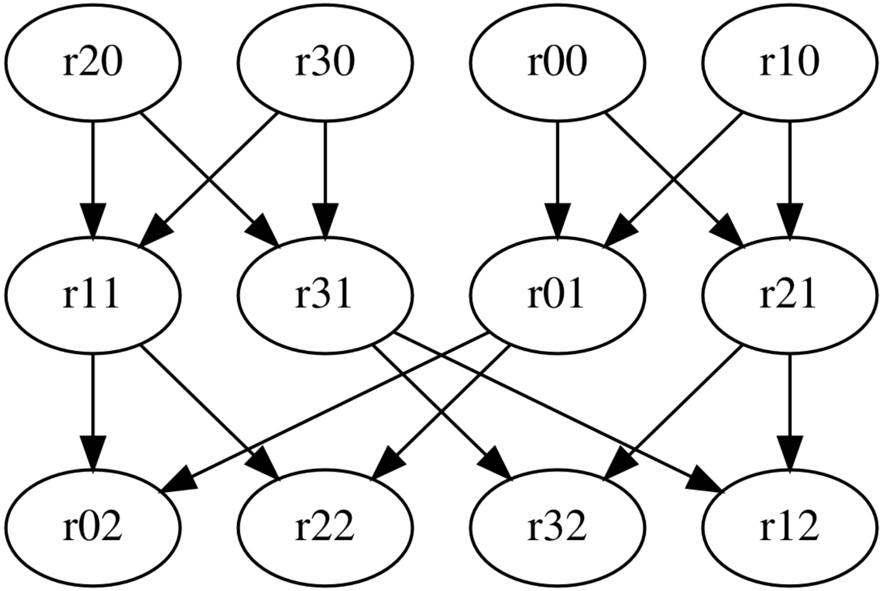
# Challenges

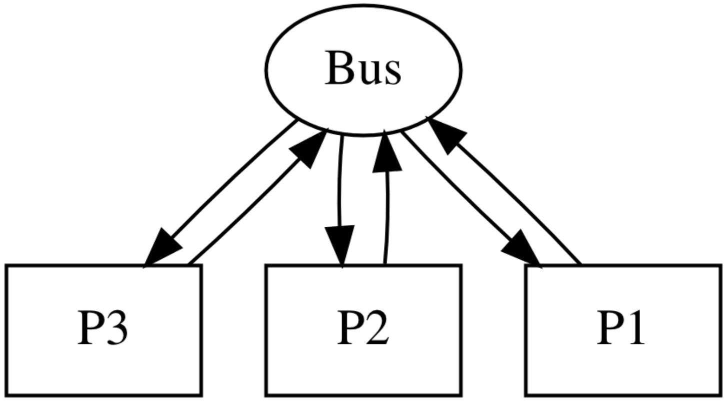
- Identify and capture usefull coarse-grain information:
  - Task costs
  - message sizes
  - processing/computing elements throughputs
  - ...
- Compile and solve the problem (it can be hard depending on the number of tasks, processing and communication elements)

# Examples

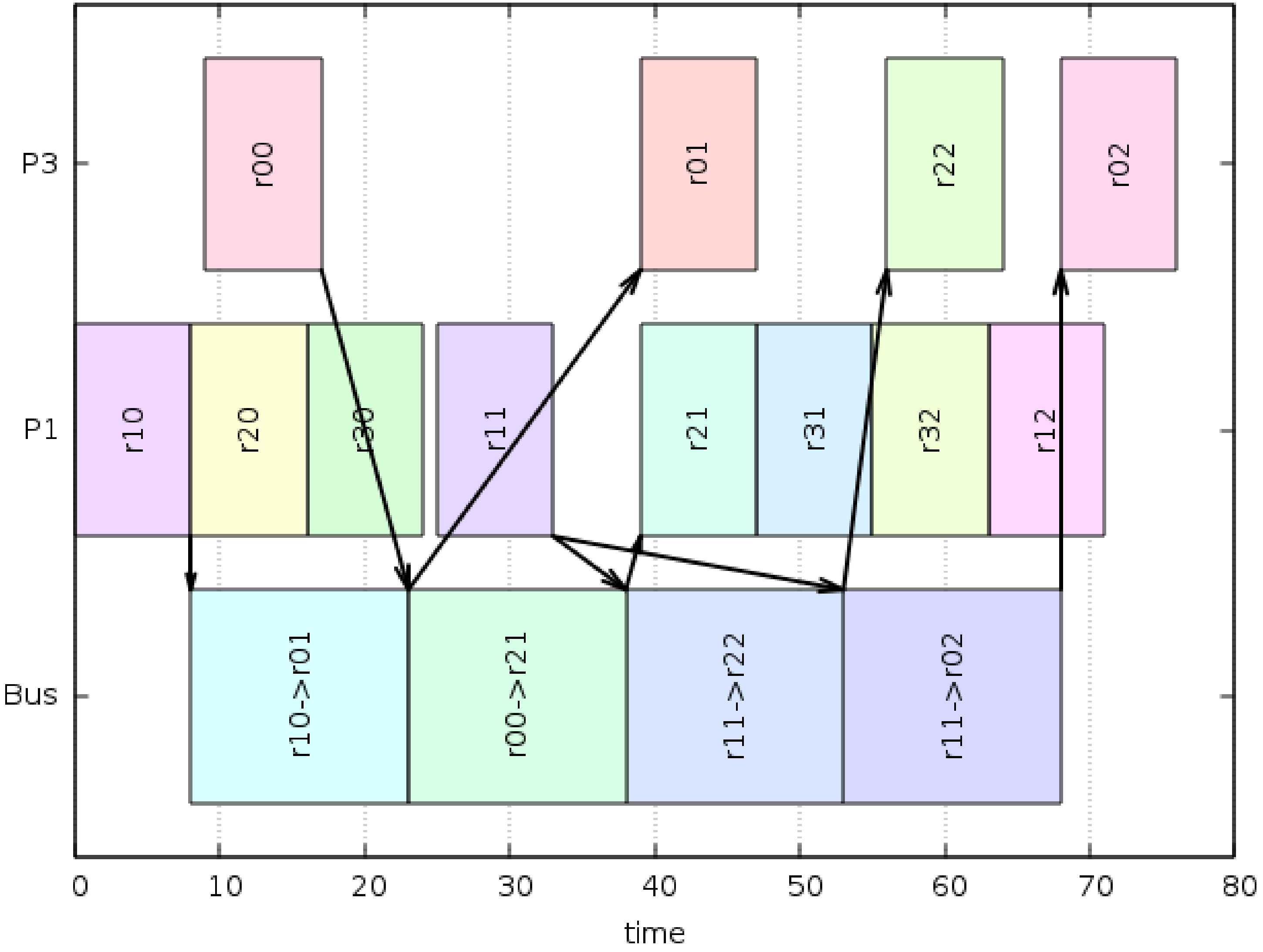
Fast Fourier Transform dataflow graph clustering and scheduling over 3 homogeneous cores connected through a Bus.

Tasks cost 10 (units of time), messages cost 15.
Messages cannot overlap. No more than 3 tasks can overlap.





# Examples

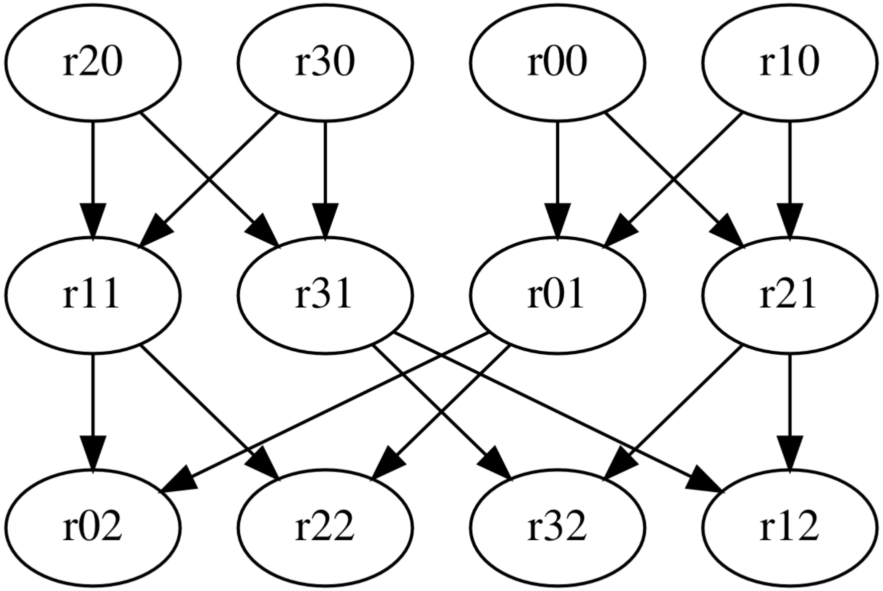


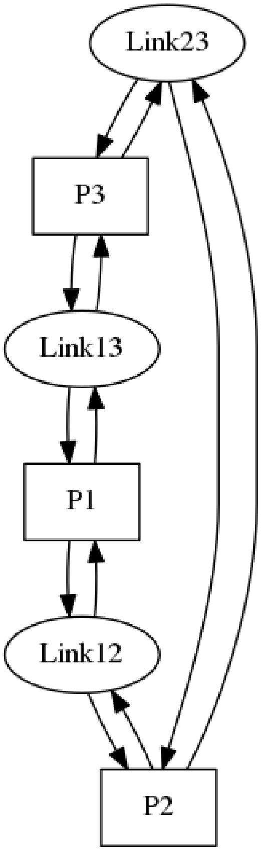
P2 not used.

# Examples

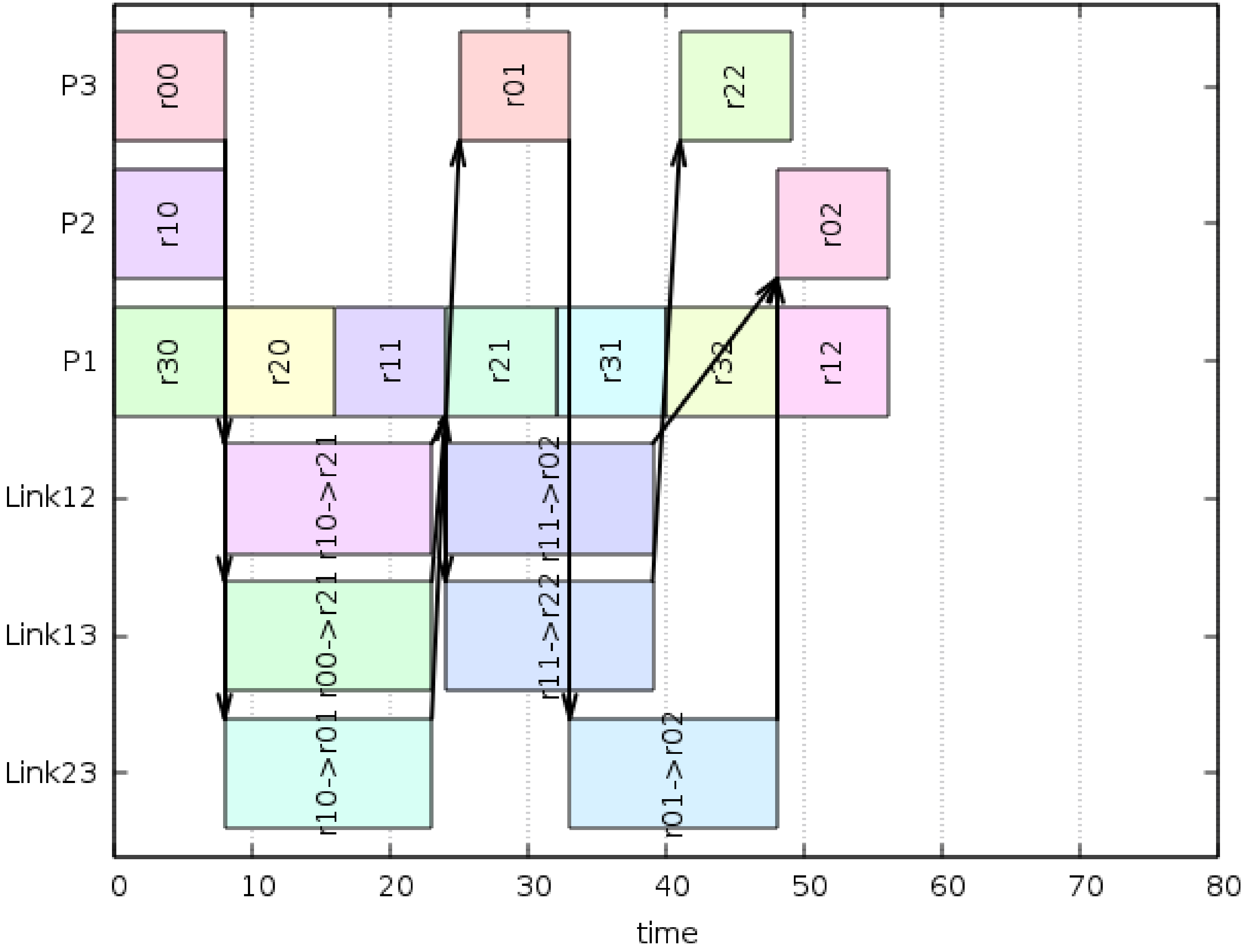
Fast Fourier Transform dataflow graph clustering and scheduling over 3 homogeneous cores connected through a Bus.

Tasks cost 10 (units of time), messages cost 15.
Some messages **can** overlap. No more than 3 tasks can overlap.





# Examples


P3
P2
P1
Link12
Link13
Link23
r00
r10
r30
r20
r11
r21
r31
r32
r12
r01
r22
r02
r10->r21
r00->r21
r10->r01
r11->r02
r11->r22
r01->r02
0
10
20
30
40
50
60
70
80
time

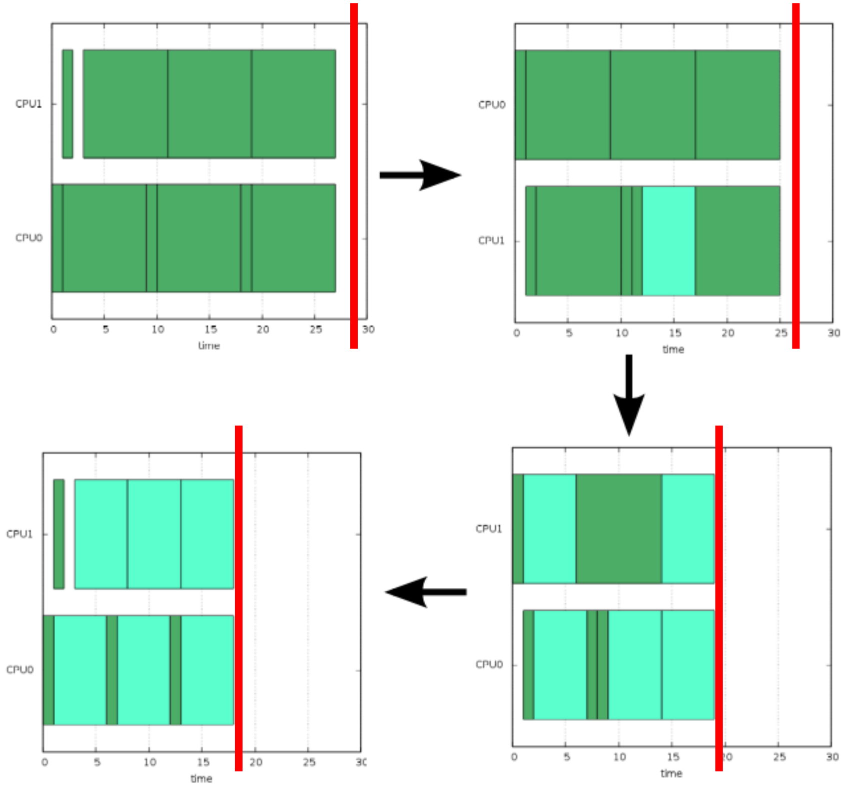
# Power opt. example

- Instead of fixed duration tasks, allow them to run different frequencies (power optimisation known as DVFS).

Ex: two cores, each one can run at two frequencies (dark=slow).

Time budget, optimise for power. Shrinking the time budget → tasks run at higher frequencies.
(or the other way around).

Power model of a processing element is ~$V^2$.f
No communication cost (either in time or power)



# Other examples and discussions

# Thank you!

Going further: contact me and/or see my poster

Questions?