An Optimal Architecture for a Multi-Standard Reconfigurable Radio: A Network Theory Re-formulation

by

V. Rodriguez, C. Moy, J. Palicot

SCEE Laboratory

IETR, Supélec, France

vr <at> ieee.org, {christophe.moy, jacques.palicot}@supelec.fr

IEEE PIMRC

Helsinki, Finland — 11-14 September, 2006
Overview of our approach: choosing between extremes

- A multi-standard radio as a graph
- A “realistic” mini-design (larger design in paper)
- The network design problem
- Available promising algorithms
- Discussion/Outlook
• To design a multistandard reconfigurable radio one must: choose between 2 extremes

• One extreme: go "Velcro": one self-contained component per standard

• Other extreme: go "primitive":
  – Use only adders, multipliers, etc.
  – provide "higher" functionality by multiple calls

• Trade-offs:
  – Velcro provides best performance, but at highest manufacturing cost (and size/weight)
  – Other extreme likely minimises cost (and size/weight) but at unacceptable performance

WHAT TO DO??
OUR APPROACH

• Find:

  BEST TRADE-OFF between PERFORMANCE and COST

• To do it:
  – build a mathematical model
  – Use suitable algorithm to solve model. In present paper, we use the:

  NETWORK DESIGN PROBLEM
• Model radio as graph of progressively simpler functional modules
• Module can be implemented in 2 ways:
  – Install a dedicated component
  – invoke (repetitively) lower level modules
• Two critical parameters per component:
  – money and time (computational delay)
• Two approaches for considering money and time:
  – Minimise weighted sum of money and time
  – Minimise monetary cost subject to “deadlines” for top modules
• The optimal design costs less (among those which respect the deadlines if applicable)
• In present work, to find the optimum we recast design as:

NETWORK DESIGN PROBLEM
Possible Dependencies

Left:
Module A needs:
EITHER B OR C

Right:
Module A needs
BOTH B & C
A Tri-standard Radio

- Standard S1
  - A1
  - B1
  - C1

- Standard S2
  - A2
  - B2
  - C2

- Standard S3
  - A3
  - B3
  - B4
  - A4
  - A5
• Want a design to support 2 main functional modules: OFDM and Equalisation
• OFDM needs fast Fourier transform (FFT)
• Equalisation (to compensate for multipath) can be implemented via
  – FIR filtering,
  OR
  – FFT (great for channels with long impulse responses)
Install or Invoke?

- **Key question: should we**
  - **install** a self-contained/dedicated component to perform a given functionality, **OR**
  - **invoke** lower level modules/components?

- A component is specified by: monetary cost and performance (execution time)

- When a lower-level component is needed several times it is called multiple times

- Choose **least expensive design** (that satisfies the “deadline” of each top module if applicable)

- Algorithm: based on network design here (exhaustive search, simulated annealing, used elsewhere)
Network Design Problem

- Want “road network” to connect “o” to each terminal $t_i$ “efficiently”
- Many possibilities. E.g.:
  - “Fastest”: 3 “direct” links
  - $O \rightarrow A$, $A$ to each $t_i$
  - $O \rightarrow A + A \rightarrow t_3 + A \rightarrow B + B \rightarrow t_1 + B \rightarrow t_2$
- Good algorithms are already available in the literature
Network Design and OUR Problem

- "Terminals" are top modules ("standards")
- "Reaching $t_i$ from $o$" means we can support top module $t_i$.
- "Building a road" from $o$ to $A$ means to INSTALL a self-contained component for $A$
- Generally, a link costs money to build and takes time to travel
- A "pre-built link" is a known algorithms that cost no money (but takes time)
The “network design” view. Solid arcs denote “pre-built” (free) “algorithmic roads” that cost time but not money.
Solid arcs denote “pre-built” “algorithmic roads” that cost no money, but do take time.
E.g., build o→Butterfly “road”, and “travel free” to FFT, and from FFT, also free to each destination.
Available Algorithms

• Bi-criteria (“cost-distance”) network-design algorithms fit well with our formulation (distance → execution time)

• Minimise weighted sum of money and time:

• Minimise cost subject to time constraint:
• To find an architecture for a multi-standard reconfigurable radio that minimises cost while considering performance objectives we model the radio as a graph of progressively simpler modules.

• KEY: install (a component) or invoke (simpler modules)?

• Easier to visualise “components” as “chips”, but approach is quite general: If DSP-based design, view “component” as “object” (object-oriented progr.). But introduce in the analysis the price-performance trade-off of the processor itself.

• To search efficiently the solution space, presently we convert graph to “NETWORK DESIGN PROBLEM”.

• A simple but realistic “mini-design” illustrates our approach.

• Available, promising “bi-criteria” algorithms have been cited.

• Even an imperfect graph-network mapping may yield a design close to “true” optimum.
Ongoing/Future Work

• Rebuilding the hypergraph of design choices. Researchers seek:
  – new operators (modules) common to several communication “blocks”
  – to replace time-domain with new frequency-domain algorithms (which would add arcs pointing to FFT)
  – to include complete communication standards in the graph, and track their evolution

• Consideration of:
  – Price/performance trade-off of DSP itself
  – multiple instances of same component (butterfly, FFT, etc) to reflect market choices
  – time needed to re-configure the radio while switching standards
  – "travel time" of signals from a component to another
  – possible contention among high level modules for the service of the same lower-level module (which may be critical if the radio needs to support simultaneous operation over several standards)
THANK YOU !!

Questions?

www.rennes.supelec.fr/ren/rd/scee/

Thanks to “Région Bretagne”, France