2004 Hardware Project Presentation

Peter Ho, Samir Omar, Nasser Esmaili,
Annie Guo, Saeid Nooshabadi,
Sri Parameswaran, Oliver Diessel

Proposed 2004 4th Year (Hardware) Thesis Projects

- Check "Thesis Topics" link at www.cse.unsw.edu.au/~cs4910
  - Username: student2003 Password: topic2003
  - Contains about 780 topics, not all of which are current
- Look for a thesis supervisor in an area that excites you
  - Want to be motivated to do your best
  - Want to show what you can do
- Feel free to propose your own project
  - Approach staff who work in a related area

Peter Ho

Nasser Esmaili

- Design and implementation of FPGA system:
  - Uses USB connection |
  - Uses Parallel connection |→ For programming
  - Uses JTAG connection |
- FPGA: VIRTEX PRO II
- Extensions:
  - Interfacing EDO-RAM (at least 8 Mb DIMMs)
  - Interfacing virtual I/O using USB expansion and Java programs
  - Interfacing Patch Boards (2 boards)

Samir Omar

Mobile Phone Design

Embedded Systems & Telecommunications

Mobile Phone Design **2003**

PIC, off line functions (phone book, messages, tones, settings, read/write SIM card)

VHDL version .. Reduced

Challenges / Future work **2004**

More functions – Full VHDL

Integration with other devices for different applications.

Embedded Web Server

Embedded Systems & Telecommunications

Embedded Web Server **2003**

PIC, Rabbit, Z8 Encore - Ethernet

VHDL

Challenges / Future work **2004**

More functions – Full VHDL

Integration with other devices (sensors)for different applications.

Wireless LAN Device

Embedded Systems & Telecommunications

Wireless LAN Device **2003**

PIC, or any.. (Z8 Encore), WLAN Card

VHDL … PCMCIA, USB, Ethernet, Serial,..

Challenges / Future work **2004**

GPS Integration, Other sensors. Full VHDL

Ad hoc Network. Sensor Network.

More Smarter

Annie Guo

Embedded Systems Design Projects

Hui Annie Guo

Introduction

- What is an embedded system?
  - Any (computer) system hidden in a product to make the product smart
  - Consists of hardware and software components that interact to perform a set of operations/functions in its host environment

Introduction

- List of embedded applications. And this can go on and on
  - Aerospace systems
  - Washing machines
  - Automobile control systems
  - Multimedia systems
  - Telecommunications
  - Industrial controllers

Introduction

- Issues in embedded systems design
  - Performance
  - Power consumption
  - Cost
  - Testability
  - Reliability
  - Etc.

Project 1: System Level Design

- Project description
  - An embedded system normally consists of a set of hardware and software components. Traditionally, they are implemented using multiple integrated circuits.
  - With the increasing complexity of integrated circuits, it is possible to implement a complete embedded system on a single integrated circuit – system-on-a-chip (SoC)

Project 1: System Level Design

- Project description (continued)
  - Growing gap between chip capacity and design productivity
    - Chip capacitor – growth rate 58%
    - Productivity – growth rate 21%
  - An efficient design methodology and good design tool are necessary

Project 1: System Level Design

- Project description (continued)
  - This project is focused on system level design and verification for embedded systems
- Prerequisites: computer architecture, knowledge of system design, experience with or willing to learn UML development tools.

Project 2: Processor Power Reduction

- Project description
  - Energy efficiency is one of important issues in embedded systems design
  - Typical target architectures for complex embedded systems consist of one or more processors, memories, and some custom designs
  - Processors are normally working horses in the system. Improving the power efficiency of processors is critical

Project 2: Processor Power Reduction

- Project description (continued)
  - This project is aimed at some design strategies to reduce power consumption in the processor
- Prerequisites: computer architecture, VHDL programming, experience with or willing to learn logic synthesis tools.

Project 3: Memory Power Reduction

- Project description
  - This project is aimed at some design strategies to reduce power consumption in the memory
- Prerequisites: computer architecture, VHDL programming, experience with or willing to learn logic synthesis tools.

Project 4: Interface Design

- Project description
  - This project is aimed at some design strategies for interfaces in embedded systems
- Prerequisites: computer architecture, VHDL programming, experience with or willing to learn logic synthesis tools.

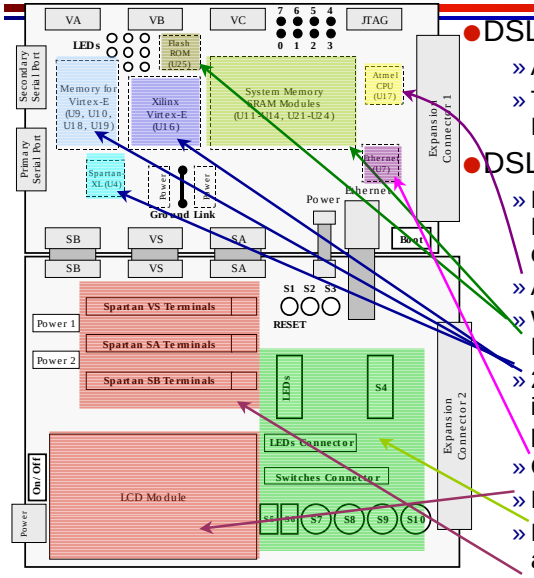
Project 5: Communication Architecture Design

- Project description
  - This project is aimed at some communication architectures for embedded systems
- Prerequisites: computer architecture, networking, good programming skills for simulation system development.

Sri Parameswaran

Saeid Nooshabadi

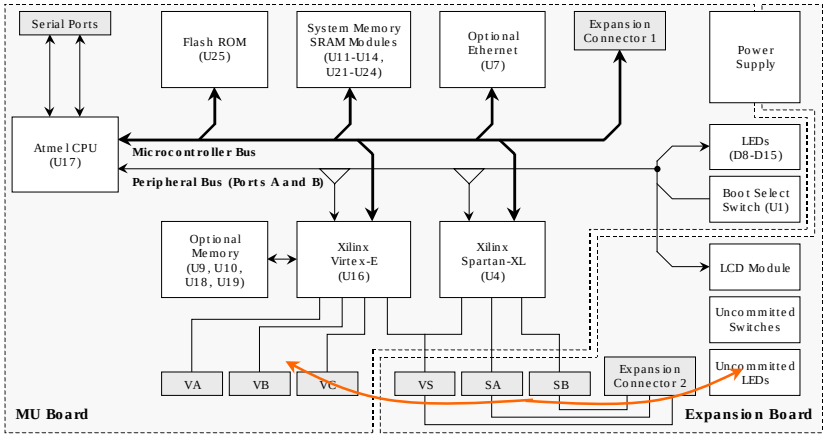
Digital Systems Laboratory Hardware



Two PCBs

- DSL Board is
  - A Board for 21st Century
  - The State-of-the-Art Development Board
- DSL Board Contains:
  - Designed by the University of Manchester with lots of collaboration from UNSW
  - An ARM Microcontroller
  - With 2 MB of Flash and up to 4 MB of SRAM Memory
  - 2 Xilinx FPGAs for extended interfacing and specialised co-processors
  - Optional Ethernet chip
  - LCD Module
  - Lots of uncommitted Switches and LEDs
  - Terminal connector to FPGAs

DSLMU Hardware Block Diagram



Projects with Digital Systems Lab Board

- Project 1: Design of a Vector Floating Point Co-Processor for ARM Core:
- Aim: The Aim of this project is to built Floating Point Vector Processor On the Vertex FPGA.
- Degree of Difficulty: Hard, and Challenging
- Ability: Number Representation, Digital System Design, DSP

Projects with Digital Systems Board

- Project 2: Design of a Fixed Point DSP for ARM Core:
- Aim: The Aim of this project is to built DSP Optimised Processor (Single Cycle MAC Processor/ Distributed Arithmetic) On the Vertex FPGA.
- Degree of Difficulty: Hard, and Challenging
- Ability: Number Representation, Digital System Design, DSP

Projects with Digital Systems Board

- Project 3: Development of a Simple Multi tasking Operating System
- Aim: The Aim of this project is Development of a Simple Multi tasking Operating System with Simple Virtual Memory Protection for on-board program monitoring and debugging.
- Degree of Difficulty: Moderate, and some Challenges
- Ability: Software Development, Basic Operating Systems
- Application: Our Undergraduate/Post Graduate Teaching
- Advantage: Make yourself Immortal!

Projects with Digital Systems Board

- Project 4: Interfacing GNU debugging tool with on-board emulator (Komodo)
- Aim: The Aim of this project is to Interface GNU debugging tool with on-board emulator program to facilitate source-level debugging and monitoring.
- Degree of Difficulty: Moderate to Hard with some Challenges
- Ability: Software Development, Basic Operating Systems
- Application: Our Undergraduate/Post Graduate Teaching
- Advantage: Make yourself Immortal!

Projects with Digital Systems Board

- Project 5: Porting of uCliux to DSL Board
- Aim: The Aim of this project is to port real-time Operating System uClinx to DSL Board.
- Degree of Difficulty: Hard, and Challenging
- Ability: Software Development, Basic Operating Systems
- Start: http://www.uclinux.org

Projects with Digital Systems Board

- Project 6: Design of an embedded Internet enabled device for remote control and monitoring
- Aim: The aim of this project is to design an embedded interface device using programmbale microcontrollers and FPGAs, to wireless devices in one hand and modem/lan on the other hand. The unit collects the data through the wireless picodevices and in turn transfers the data via modem or lan to a remote server via TCP/IP stacks. It should also be possible to control the picodevices remotely. The challenge is to build the various software layers, on a realtime operating system like eCos to do the task.
- Degree of Difficulty: Hard, and Challenging
- Ability: Software Development, Basic Operating Systems, basic Hardware Interfacing

Projects with Digital Systems Board

- Project 7: Design of an embedded Internet enabled device for remote control and monitoring
- Aim: The aim of this project is to design an embedded interface device using programmable microcontrollers and FPGAs, to control and monitor the viewing of cable TV channels.This embedded unit controls a TV tuner, in a real time fashion, based on the control information it receives remotely. The unit is connected to the internet via a cable modem. It can be controlled and monitored via a remote device (a PC). The challenge is to build the various software layers, on a realtime operating system like eCos to do the task.
- Degree of Difficulty: Hard, and Challenging
- Ability: Software Development, Basic Operating Systems, basic Hardware Interfacing

Projects with Digital Systems Board

- Project 8: PS/2 and USB Controller For DSL Board
- Aim: The aim of this project is to design an PS/2 and USB Controller using the on-board FPGA chips.
- Degree of Difficulty: Hard, and Challenging
- Ability: Hardware Development, Basic Software development

Projects with Digital Systems Board

- Project 9: Frame Grabber Device for CMOS Digital Camera
- Aim: The aim of this project is to build a high resolution web camera using a Kodak KAC-1310 CMOS image sensor. The pixel data would buffered in SRAM /SDRAM, compressed and uploaded through the Internet interface for display on a PC.
- Degree of Difficulty: Very Hard, and Challenging
- Ability: DSP, Hardware Development, Basic Software development

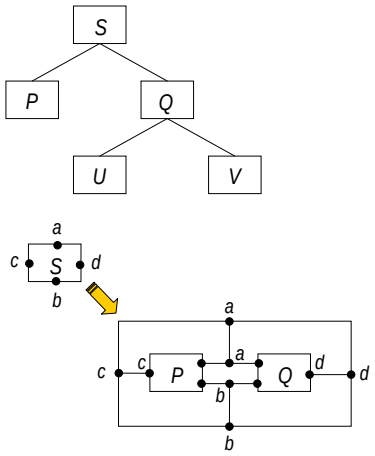
Projects with Digital Systems Board

- Project 10: Audio Signal Processing
- Aim: The aim of this project is interface a codec to a DSP Audio Signal Processor/ Compressor built on the on-Board FPGAs
- Degree of Difficulty: Moderate to Hard, and Challenging
- Ability: DSP, Hardware Development, Basic Software development

Oliver Diessel (Topics contain “OFD”)

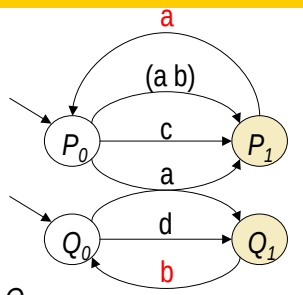
- Circal interpreter & engine
- FPGA-based robot controller design
- FPGA-based Ant Colony Optimization problems
- Traceable copy technology
- Smooth, digitized handwriting

Circal basics



- The Circal process algebra supports hierarchical, modular, and constructive description of interacting processes
- *Processes* are behavioural objects that interact based on the occurrence of events

Structural modelling in Circal



- Structural operators
  - Composition
    - Evolve on shared events only when each is independently able to

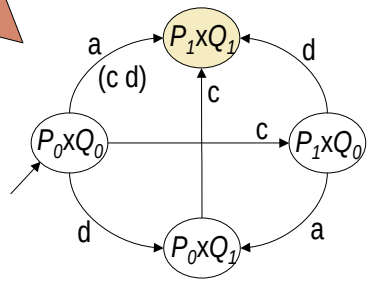
$S \leftarrow P_0 * Q_0$

$P_0 * Q_0 \leftarrow$ c $P_1 * Q_0$ + d $P_0 * Q_1$ + a $P_1 * Q_1$ + (c d) $P_1 * Q_1$

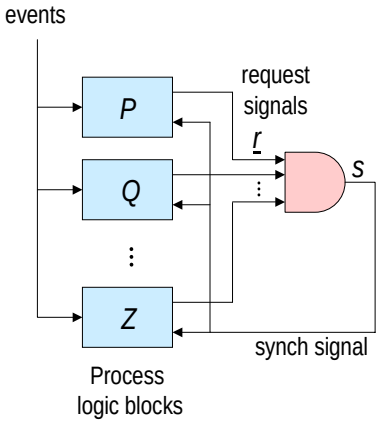
$P_1 * Q_0 \leftarrow$ a $P_0 * Q_1$ + d $P_1 * Q_1$

$P_0 * Q_1 \leftarrow$ c $P_1 * Q_1$

$P_1 * Q_1 \leftarrow \Delta$ (DEADLOCK)

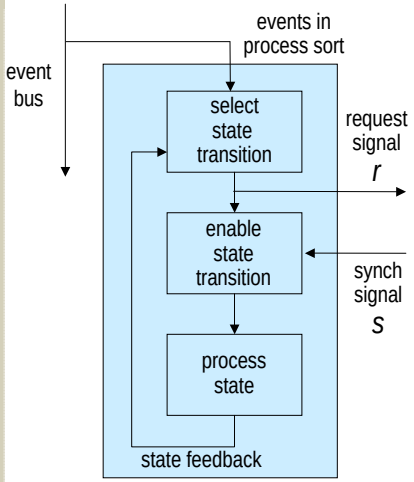


Circuit realization of Circal



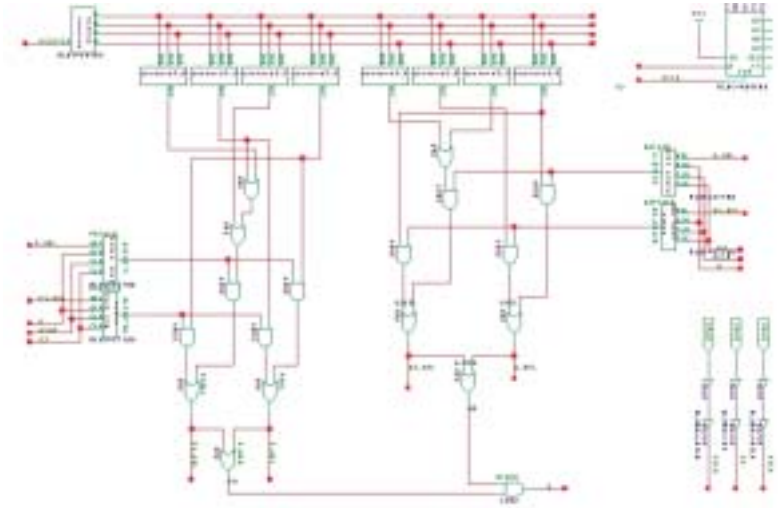
- At the system level, a Circal specification is realized as an interconnection of independent, concurrently active *process logic blocks* and *synchronisation logic*

Circuit realization of Circal (cont)

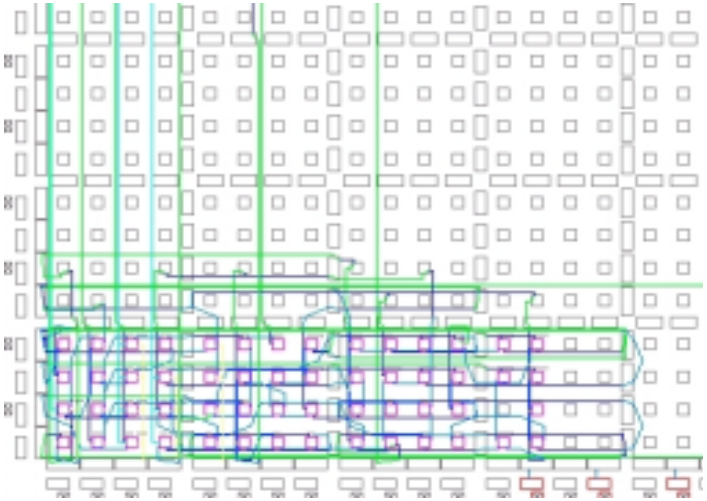


- Each process logic block implements the behaviour specified by its process definitions
- Of course the Circal composition law constrains process state transitions to those that are globally acceptable

Schematic for *P***Q*



Automatic place & route of flat design



Module Generators

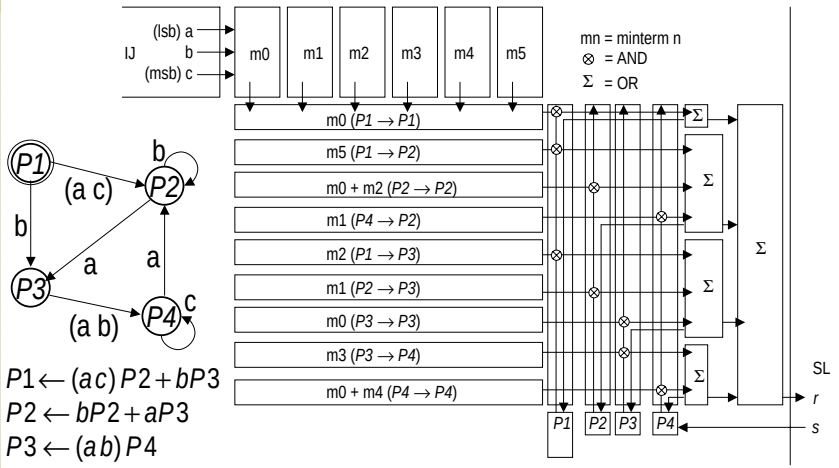
9. State registers:

   SR(tlc, tlr, tf);

   - implements initial or non-initial state using selector from N and enable from E

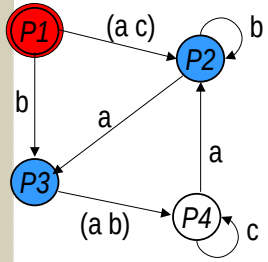
   - specified by coords of top left corner, height, and state type flag

Static circuit implementation



$P1 \leftarrow (ac)\,P2 + bP3$
$P2 \leftarrow bP2 + aP3$
$P3 \leftarrow (ab)\,P4$
$P4 \leftarrow cP4 + aP2$

Circuit modelling & partitioning



$P1 \leftarrow (ac)P2 + bP3$

$P2 \leftarrow bP2 + aP3$

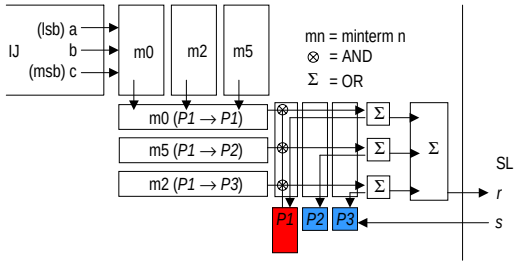
$P3 \leftarrow (ab)P4$

$P4 \leftarrow cP4 + aP2$

- Processes are modelled as state transition graphs and processes are partitioned according to their definitions
- Initially, the interpreter implements a sub-graph rooted at the initial state
- Nodes are included breadth-first until it is not possible to fit the transition logic for the next state

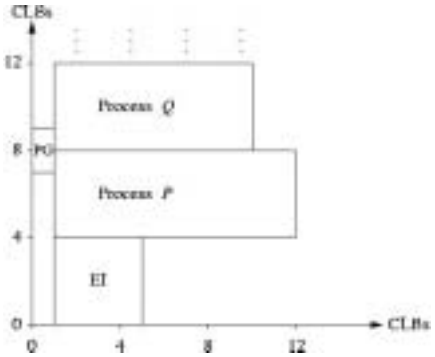
Example

- Suppose the array area for process P can only accommodate the behaviour for state P1
- To determine which transition occurred, *boundary state* registers for P2 and P3 are needed as well



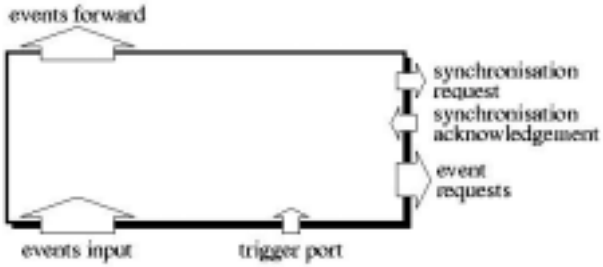
Generalized processes

- Work done by Jérémie Detrey to implement *hierarchy, abstraction* and *process creation*
- Developed on Wildcard XCV300 implementation of compiler



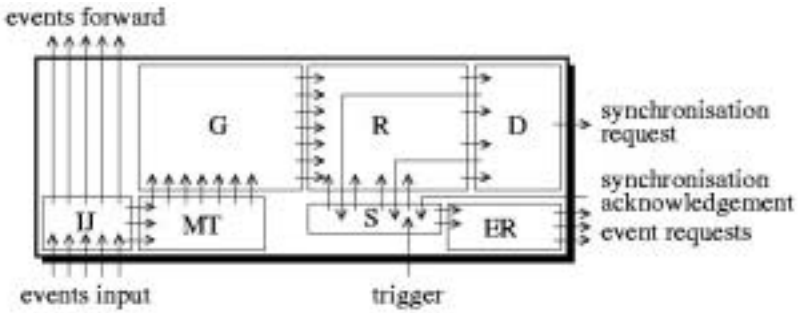
Abstract process interface

- For the variety of process blocks required, a *common process interface* has been defined

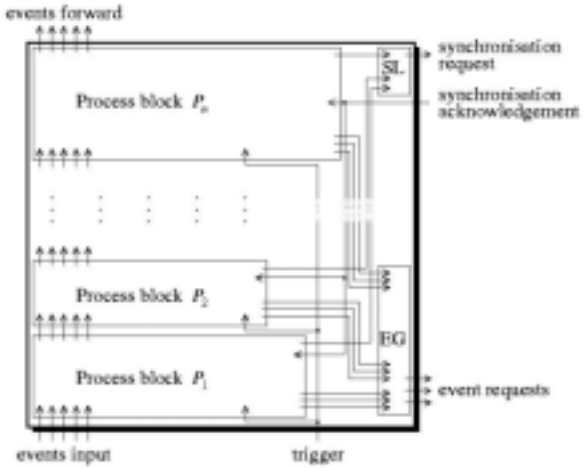


FSM process

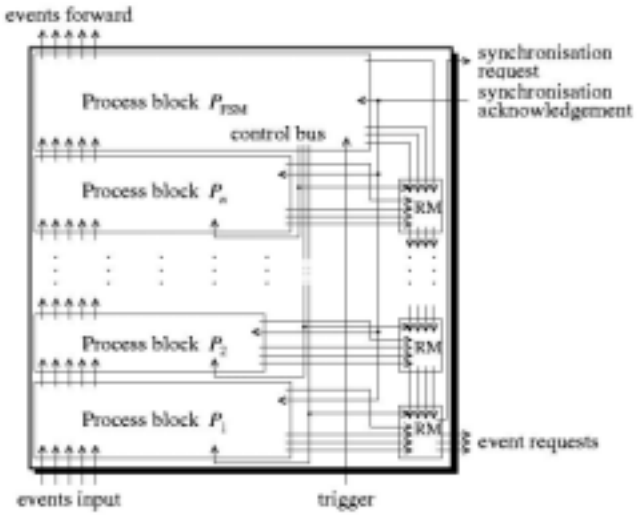
- As before, only allows *choice* and *guarding*, but process can be switched on/off and provides for *abstracted (internal)* events



Hierarchical composition

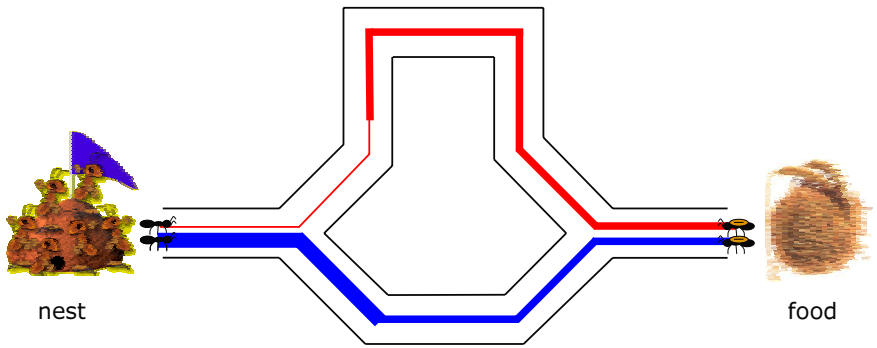


Hybrid process block layout



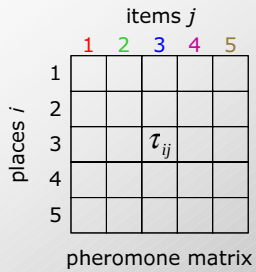
Explanation

- Ants deposit pheromone on their path
- Ants prefer paths with more pheromone



Evaporation based ACO

by Dorigo et al.: 1992/96/99



Search for good solutions represented by permutations of n items.

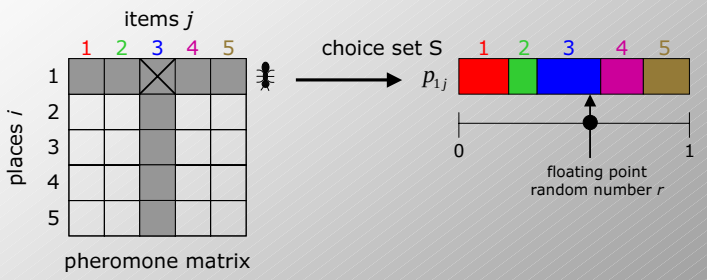
Example:
items (2, 5, 3, 1, 4)
places [1, 2, 3, 4, 5]

Solution, e.g. job sequence scheduled on a single machine

Evaporation based ACO

by Dorigo et al.: 1992/96/99

Generate Solution: sequence of local decisions

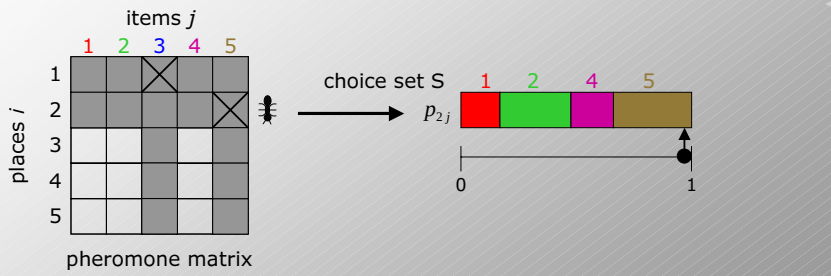


For a given place *i* an ant chooses item *j* according to $p_{ij} = \frac{\tau_{ij}}{\sum_{k \in S} \tau_{ik}}$

Evaporation based ACO

by Dorigo et al.: 1992/96/99

Generate Solution: sequence of local decisions



For a given place *i* an ant chooses item *j* according to $p_{ij} = \frac{\tau_{ij}}{\sum_{k \in S} \tau_{ik}}$

Evaporation based ACO

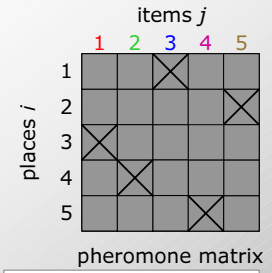
by Dorigo et al.: 1992/96/99

Generate Solution: sequence of local decisions

after *m* solutions have been generated

Pheromone Update:

- Global evaporation on all pheromone values:
  $\forall i,j \in [1,n] : \tau_{ij} \mapsto (1-\rho) \cdot \tau_{ij}$
- Increase pheromone of best solution π:
  $\forall i \in [1,n] : \tau_{i\pi(i)} \mapsto \tau_{i\pi(i)} + \Delta$



ant generated solution
(3, 5, 1, 2, 4)

repeat until stopping condition is met

PACO Architecture on FPGA