Design of Transport Triggered Architectures

by Henk Corporaal

Presentation by Tommi Salminen

Paper organization

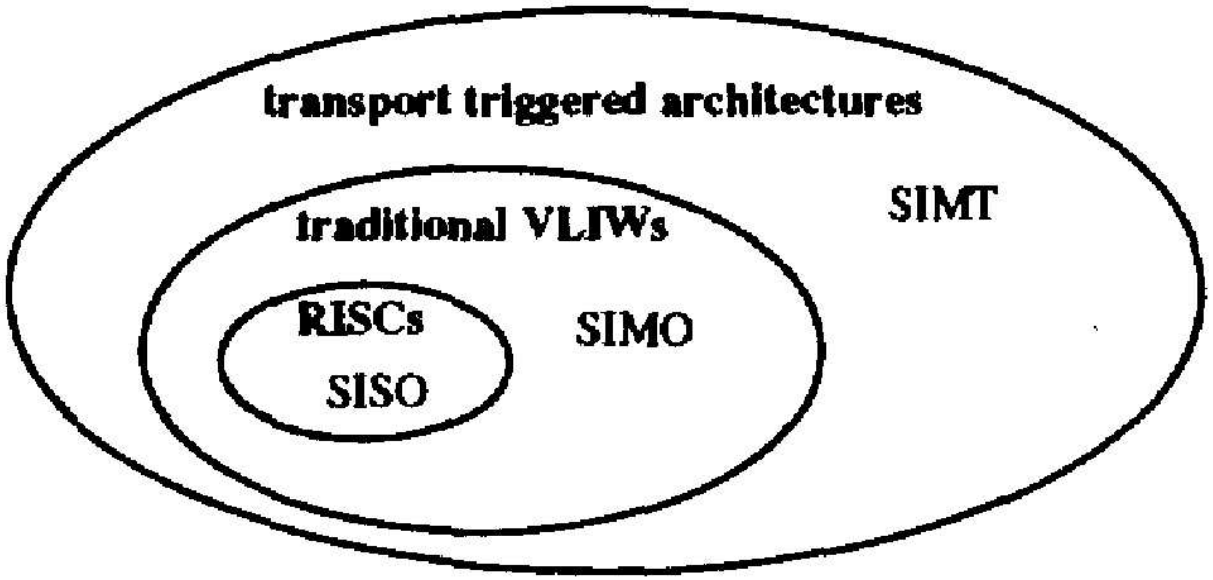
- **Concept of transport triggering**
- **MOVE32INT – a prototype TTA processor**
- **Automatic generation of arbitrary TTAs**

Why TTA

- VLIW & Superscalar architectures:
  - complex organizations
  - poor hardware utilization
  - difficult to change functionality
  - poor performance scalability

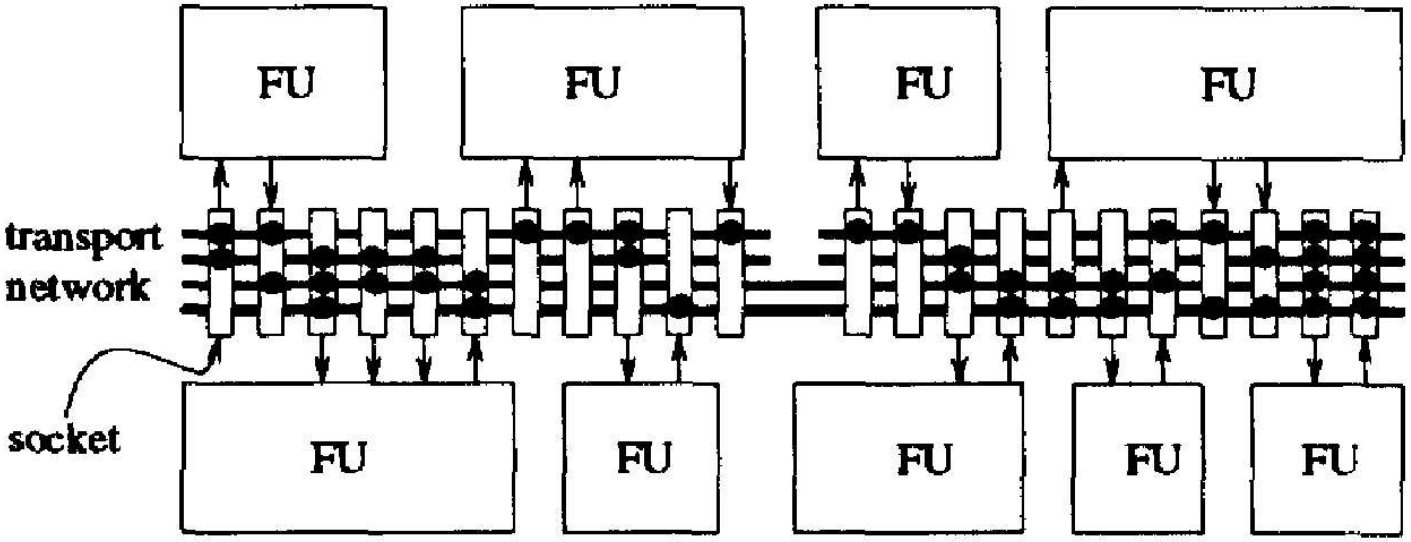
=> From operation triggered to transport triggered processors

Architecture space of long instruction word machines



- **SISO – Single instruction, single operation**
- **SIMO – Single instruction, multiple operation**
- **SIMT – Single instruction, multiple transports**

General Structure of a MOVE architecture

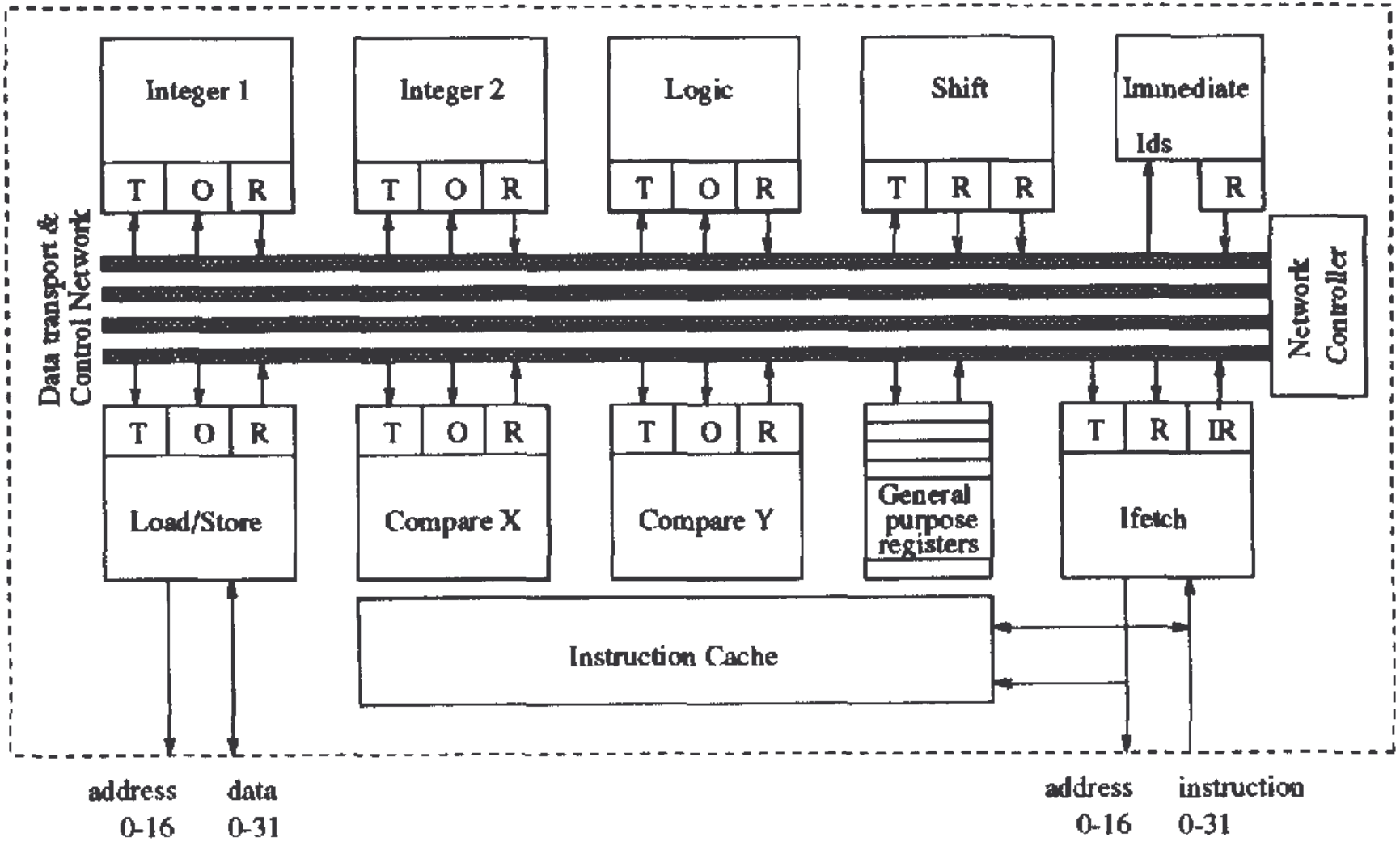


- **Function units (FUs)**
- **Sockets**
- **Buses**
- **Registers (Operation, Trigger, Result)**

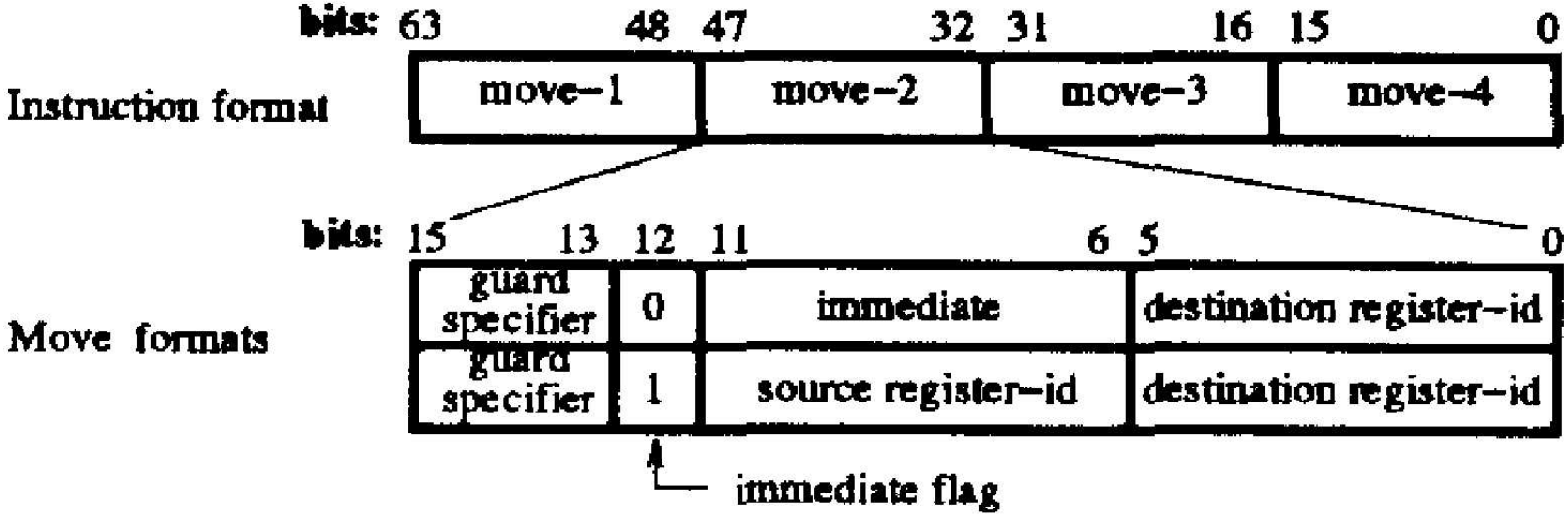
MOVE32INT

- 32-bit general purpose pipelined processor running at 80MHz
- Realized on a Sea of Gates chip of 200k transistors (50% used).

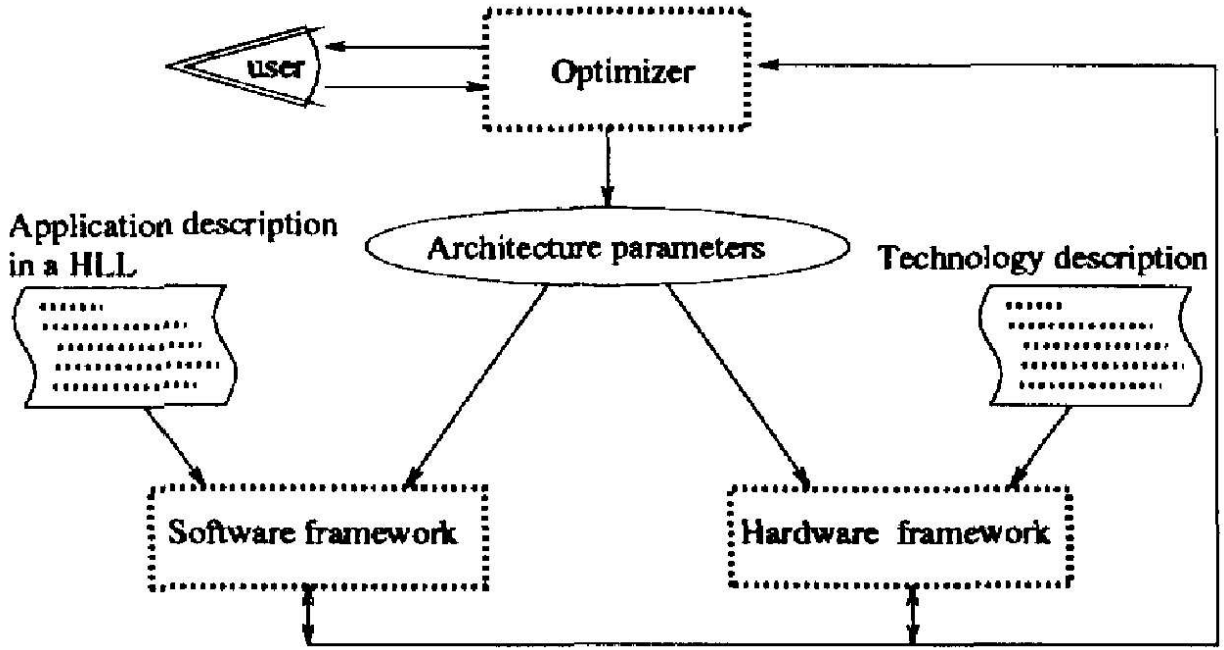
MOVE32INT block diagram



MOVE32INT instruction and *move* formats



Automated design



- In: Application description, cost and performance constraints
- Out: Chip layout, compiler for this TTA

Summary

- TTA advantages:
  - Simplicity and therefore quick design time
  - Flexibility and performance scalability (transport network and FUs orthogonal and can be optimized independently)
  - Short processor cycle time (FU superpipelining; cycle time from data transport)
  - Better scheduling (fine grain operations)
  - Efficient usage of transport capacity
  - Less register usage (values passed directly from result registers to operand and trigger registers)
- TTA disadvantages:
  - More complex code generation process (compiler now has to also schedule transports)