Networks and Applications: Are Application-Specific Networks Worth The Trouble?

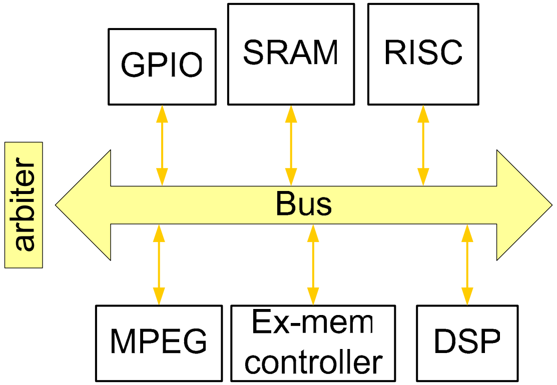
Jiang Xu and Wayne Wolf
Dept. of EE
Princeton University

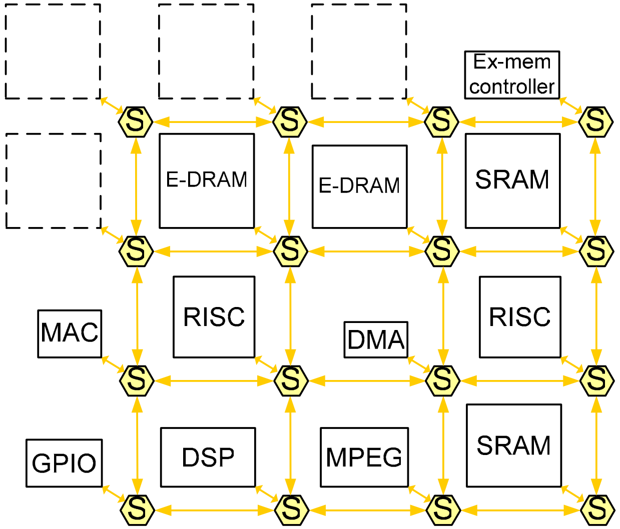
Outline

- Case studies of real applications
  - H.264 HDTV decoder system-on-chip
  - Smart camera system-on-chip
- Methodologies for application-specific NoC design
- A step back

Current on-chip communication architectures

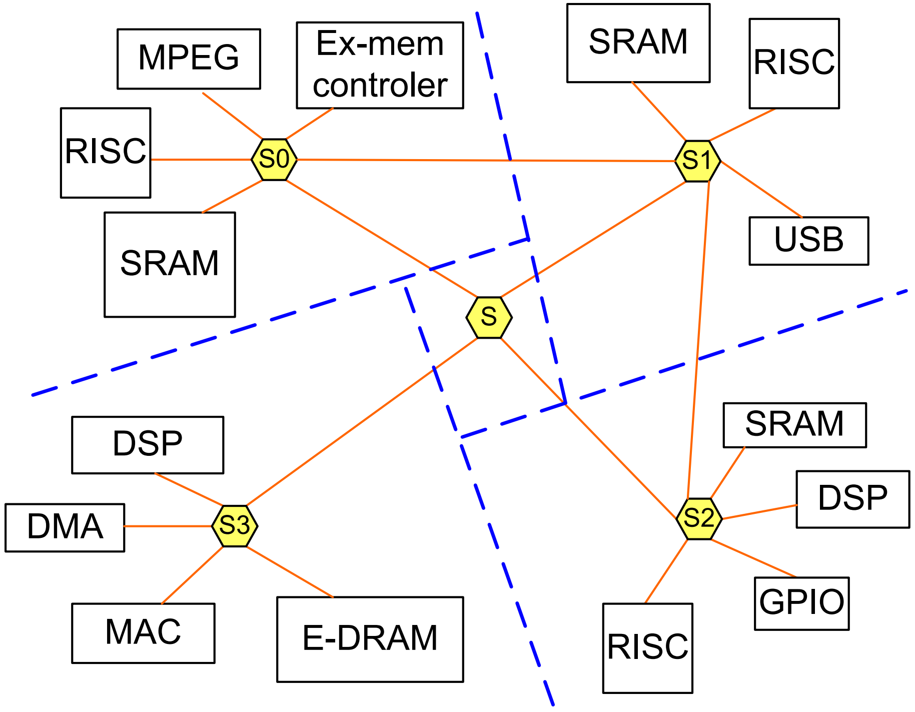
- On-chip bus
  - CoreConnect, AMBA, Wishbone, μNetwork …
  - Mature technology
  - Shared media
- Regular-topology networks-on-chip
  - RAW, CLICHÉ, Nostrum, Eclipse, aSoC, SPIN …
  - From multi-computer networks
  - Designed for general-purpose and homogenous systems





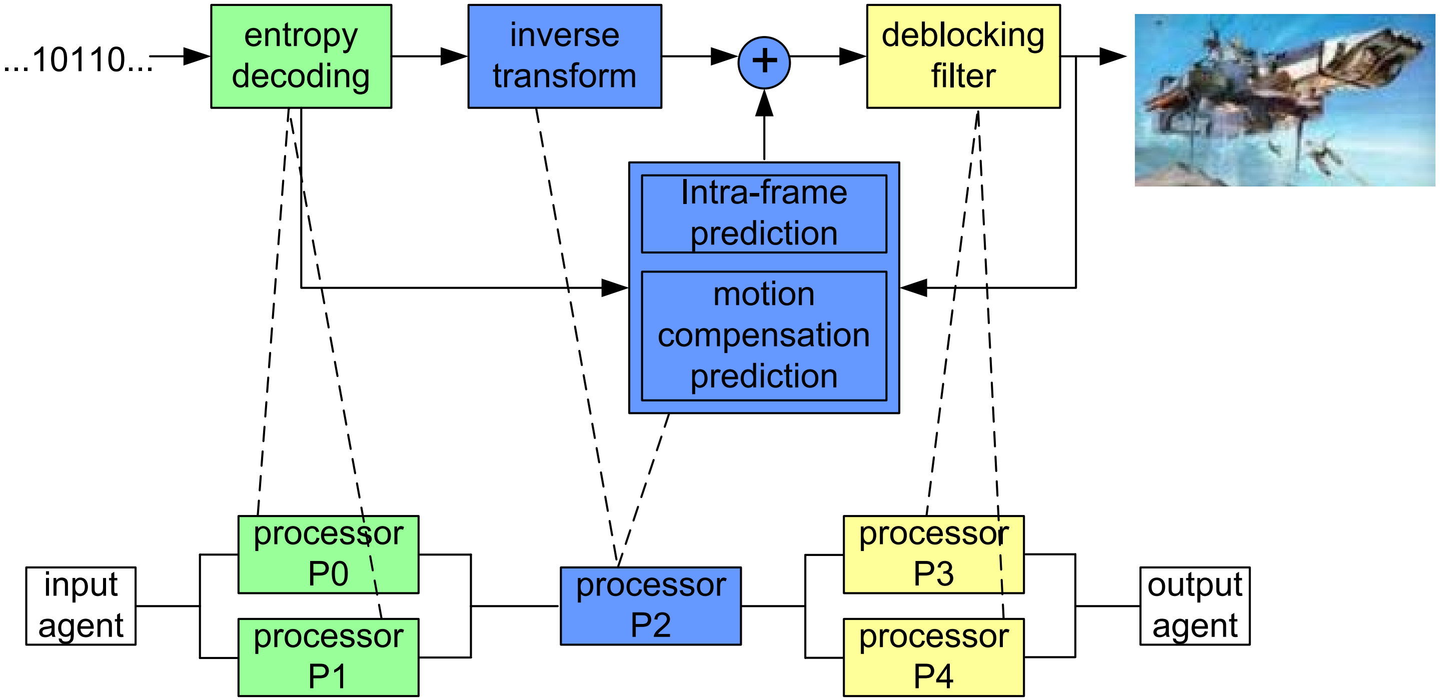
Application-Specific Networks-on-Chip

- Customized on-chip network for each application
  - ASNoC brings significant performance improvements
- Irregular-topology and hierarchical
- For both homogenous and heterogeneous systems
- Based on a network component library

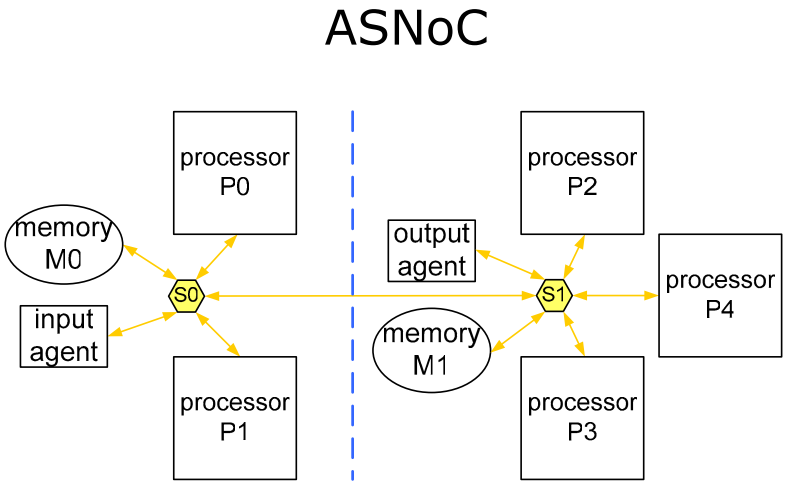


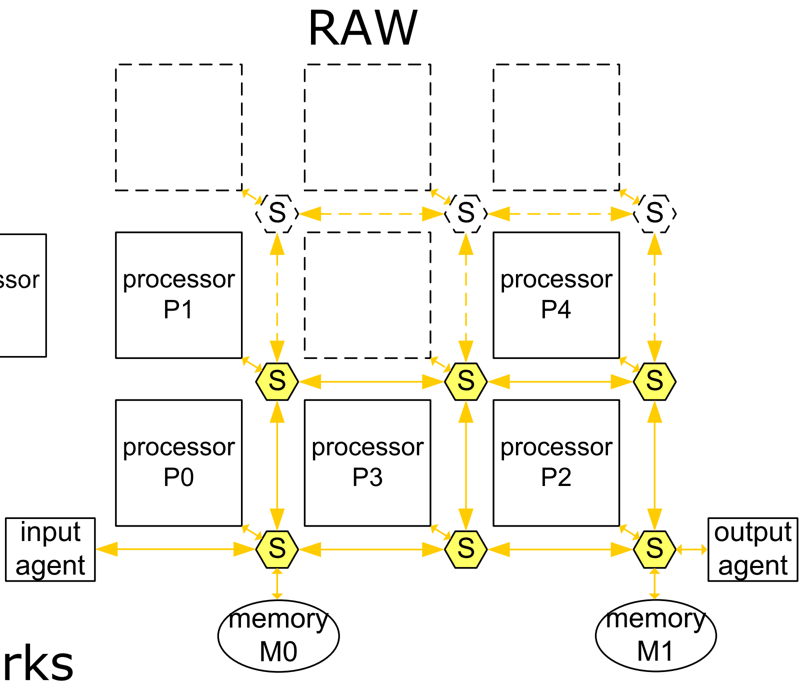
H.264 HDTV decoder system-on-chip:
Behavior model & computation architecture

- Candidate for HDTV broadcast
- High compression rate: 2X of MPEG2
- High definition: 2 million pixel/frame



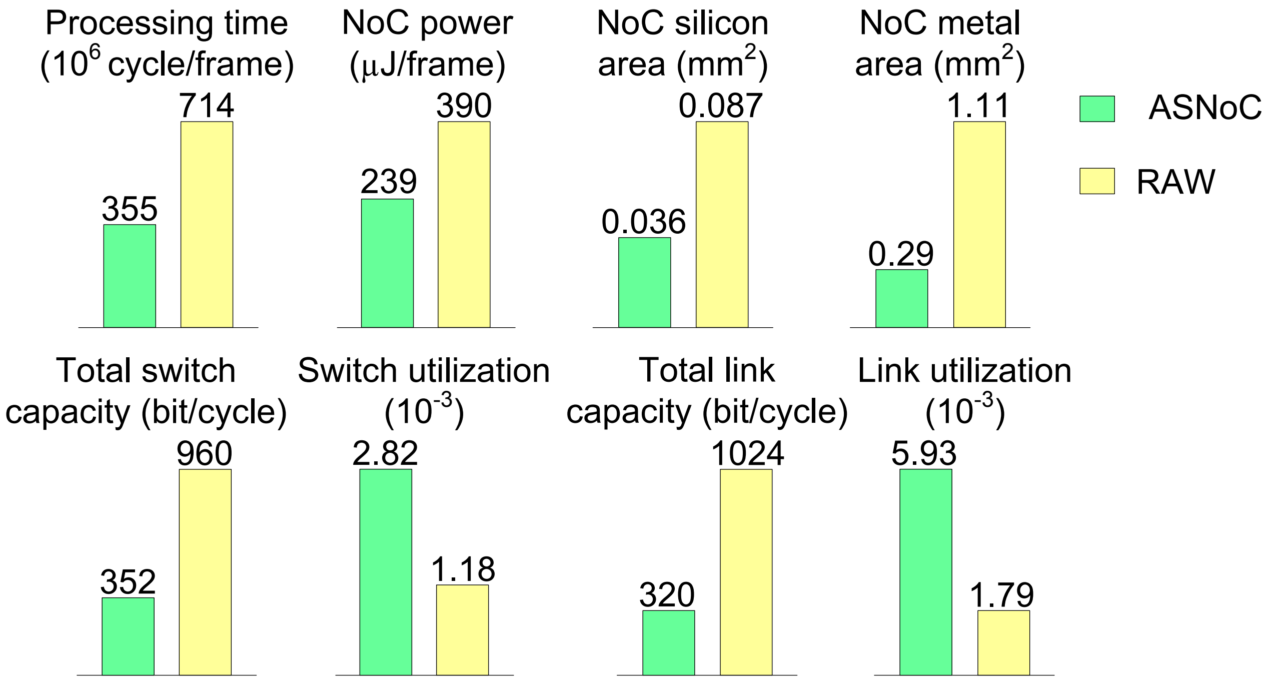
H.264 HDTV decoder system-on-chip:
RAW vs. Application-Specific Networks-on-Chip





- ASNoC has two local networks
- RAW is implemented based on its design documentation
- Positions of computation nodes are optimized
- The same group of computation nodes
- Different communication architectures
- ASNoC has less switches and links

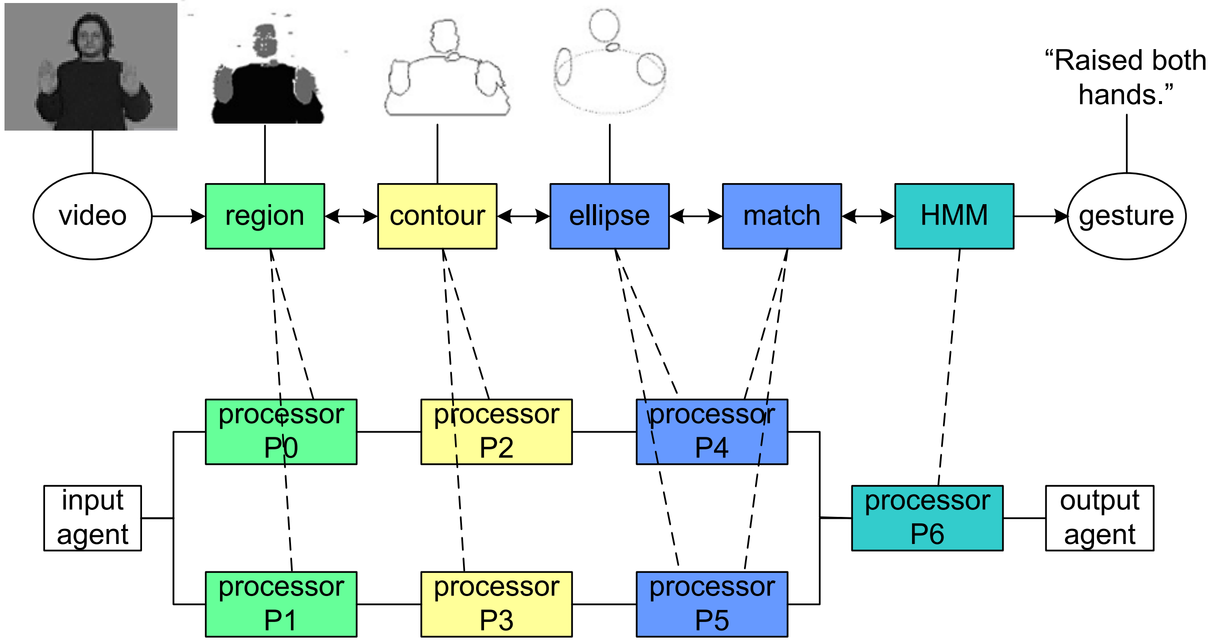
H.264 HDTV decoder system-on-chip: Results and comparison



- ○ Higher performance: 201%
- ○ Lower power: 61%
- ○ Less area: 26% metal area, 41% silicon area
- ○ Less network resource: 37% switch capacity, 31% link capacity
- ○ Higher network utilization: 239% switch utilization, 331% link utilization
- ○ ASNoC is better than regular-topology networks-on-chip

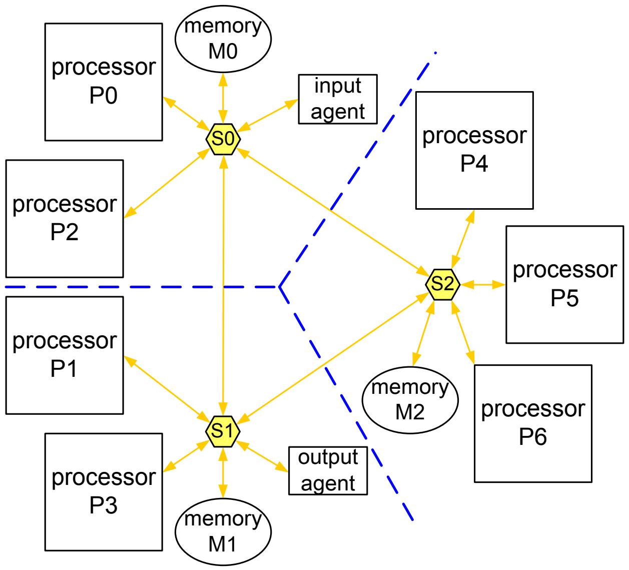
Smart Camera system-on-chip:
Behavior model & computation architecture

- Real-time gesture recognition
- 150 frame/sec
- Dual-pipeline computation architecture

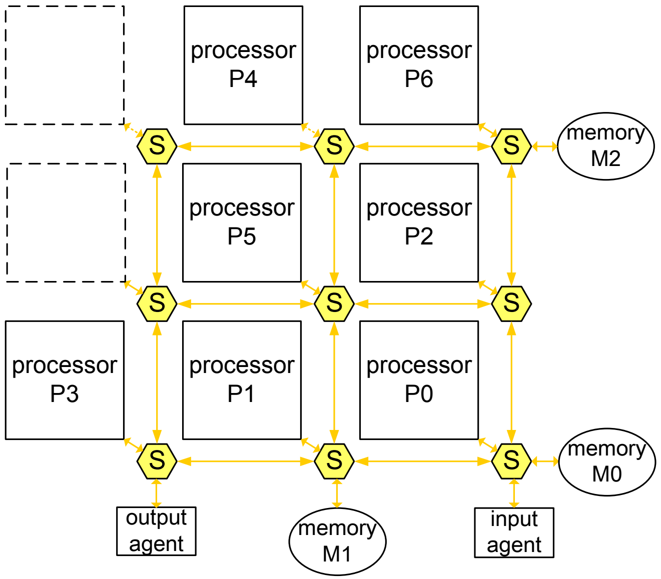


Smart Camera system-on-chip:
RAW vs. Application-specific Networks-on-Chip

ASNoC

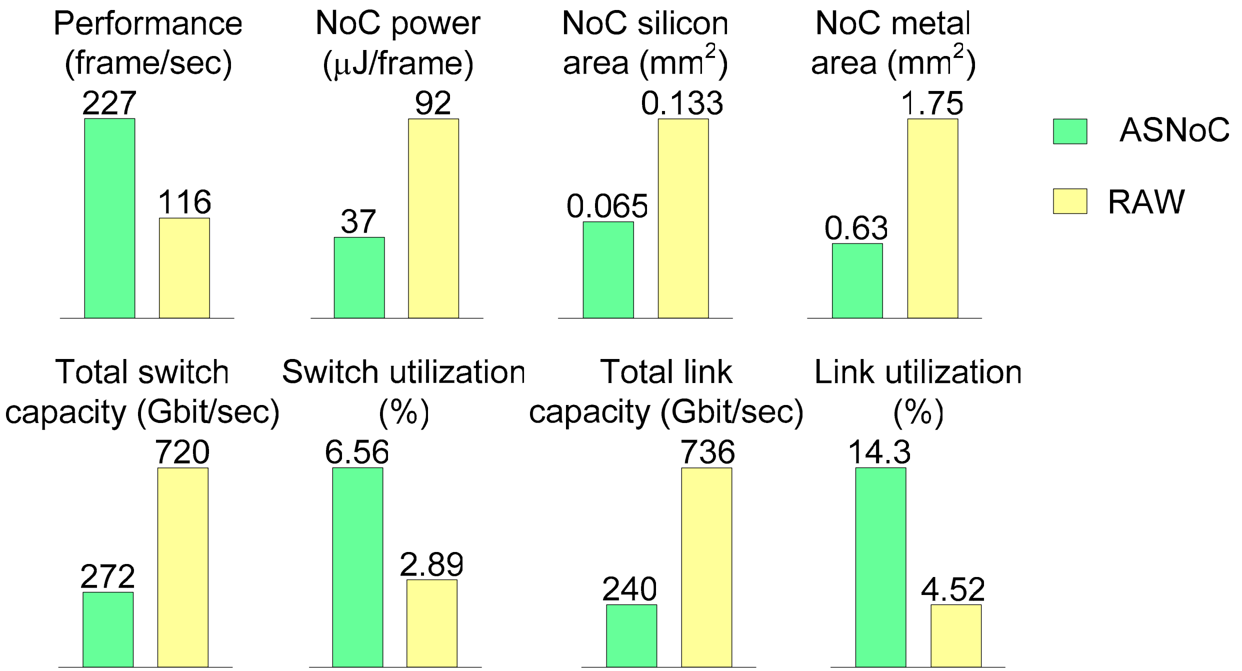


RAW



- ASNoC has three local networks
- RAW is implemented based on its design documentation
- Positions of computation nodes are optimized in RAW
- The same group of computation nodes
- Different communication architectures
- ASNoC has less switches and links

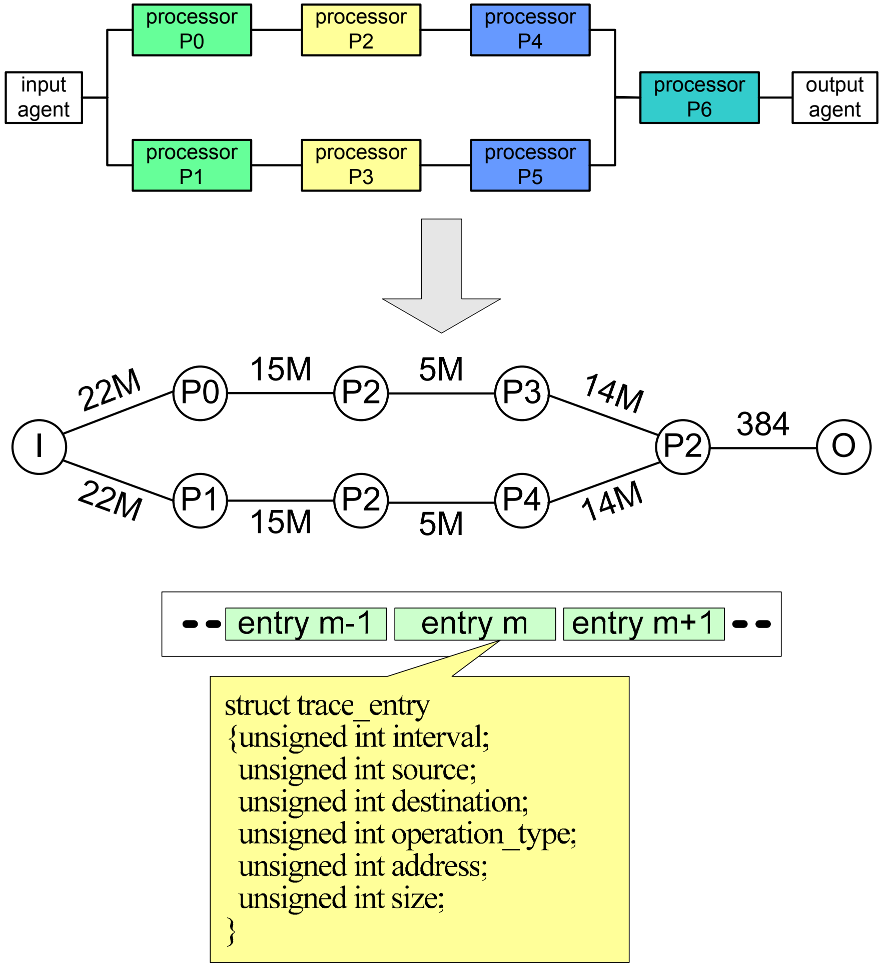
Smart Camera system-on-chip: Results and comparison



- Higher performance: 196%
- Lower power: 40%
- Less area: 36% metal area, 49% silicon area
- Less network resource: 38% switch capacity, 33% link capacity
- Higher network utilization: 227% switch utilization, 316% link utilization

Communication analysis

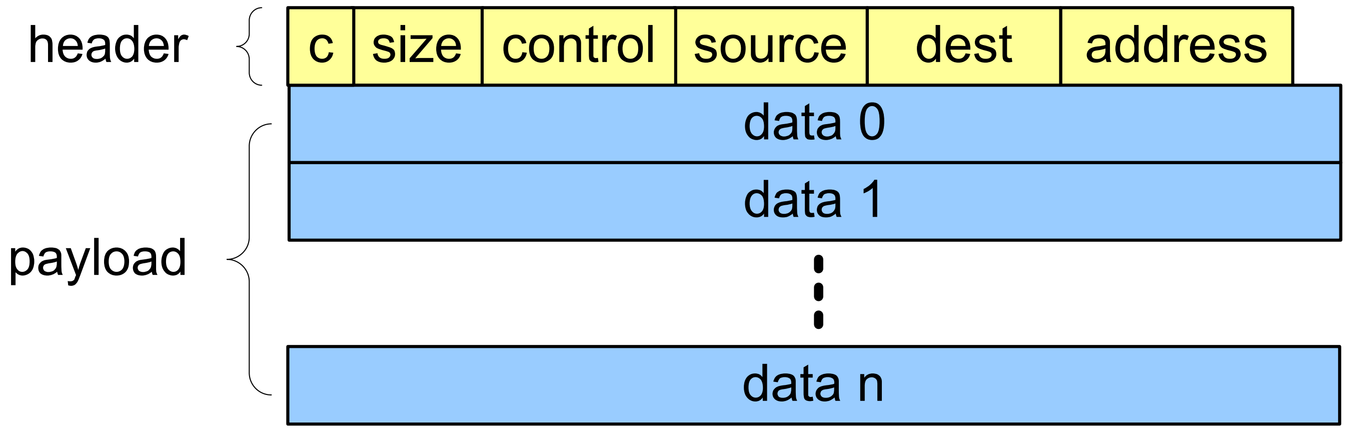
- Communication graph
  - *G=(V,E)* is weighted
  - *v*∈*V* is a computation node
  - *e*∈*E* is a connection between nodes
  - Weight *w(e)* is the average comm. traffic in a fixed period of time
- Recorded comm. trace
  - from computation architecture simulation
  - One for each node
  - Control comm. behavior of node



```
struct trace_entry
{unsigned int interval;
 unsigned int source;
 unsigned int destination;
 unsigned int operation_type;
 unsigned int address;
 unsigned int size;
}
```

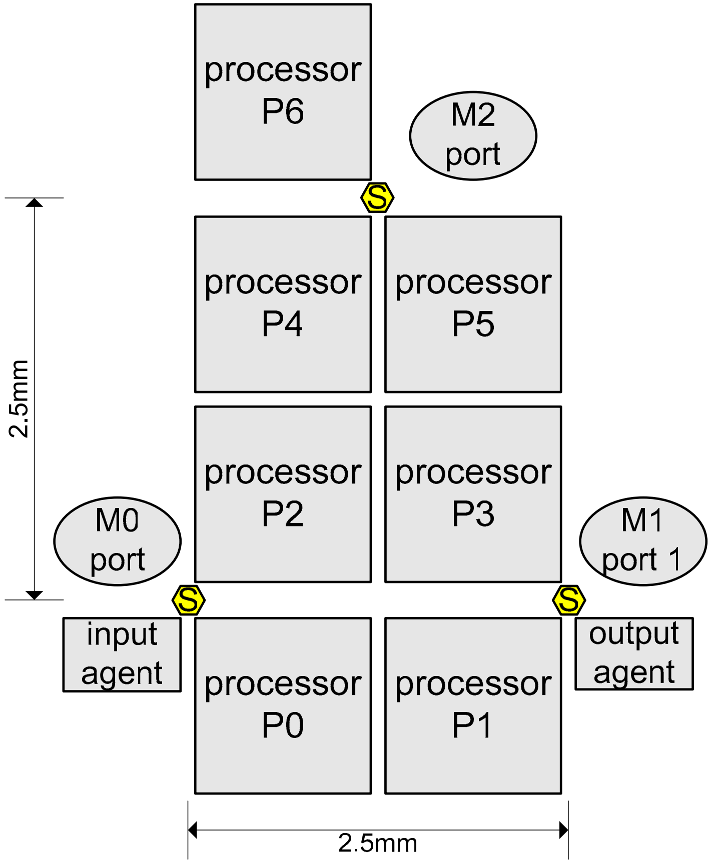
ASNoC protocol design

- Deterministic routing (adaptive routing)
- Wormhole switching (packet switching and VCT switching)
- Packet format is adjusted based on applications



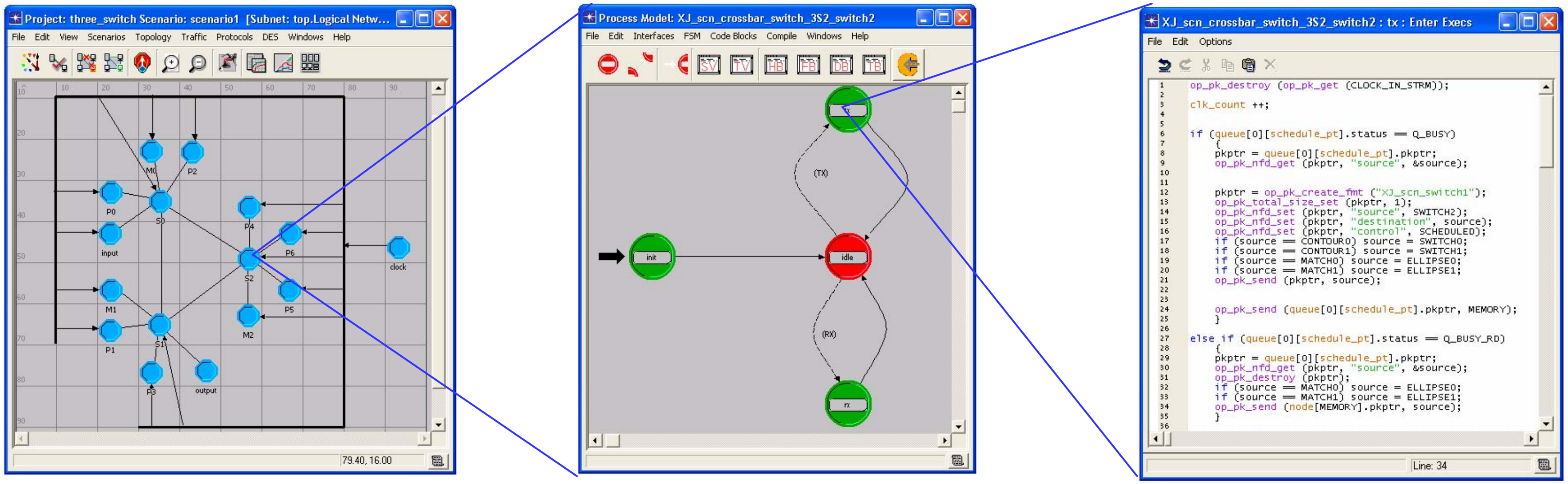
Floorplan estimation

- Slicing floorplan
  - Considering network hierarchy
  - A method similar to [Yuen03]
- High-level planning
  - not detailed placement and routing
- Used to decide link length
  - Delay for performance analysis
  - Power and area analysis



Performance analysis

- Cycle-accurate simulation
  - Based-on cycle-accurate component models in the library
- Application-level performance
  - H.264 HDTV decoder: cycle/frame, 5 billion cycles
- OPNET, a network simulator, is adapted
  - Link, transmitter, receiver, clock
- Capacity and utilization for each network component


Project: three_switch Scenario: scenario1 [Subnet: top.Logical Netw...
File Edit View Scenarios Topology Traffic Protocols DES Windows Help
M0
P2
P0
S0
P4
P6
clock
input
S2
P5
M1
P1
S1
M2
P3
output
79.40, 16.00
Process Model: XJ_scn_crossbar_switch_3S2_switch2
File Edit Interfaces FSM Code Blocks Compile Windows Help
tx
(TX)
init
idle
(RX)
rx
XJ_scn_crossbar_switch_3S2_switch2 : tx : Enter Execs
File Edit Options
Line: 34

```
op_pk_destroy (op_pk_get (CLOCK_IN_STRM));

clk_count ++;


if (queue[0][schedule_pt].status == Q_BUSY)
    {
    pkptr = queue[0][schedule_pt].pkptr;
    op_pk_nfd_get (pkptr, "source", &source);


    pkptr = op_pk_create_fmt ("XJ_scn_switch1");
    op_pk_total_size_set (pkptr, 1);
    op_pk_nfd_set (pkptr, "source", SWITCH2);
    op_pk_nfd_set (pkptr, "destination", source);
    op_pk_nfd_set (pkptr, "control", SCHEDULED);
    if (source == CONTOUR0) source = SWITCH0;
    if (source == CONTOUR1) source = SWITCH1;
    if (source == MATCH0) source = ELLIPSE0;
    if (source == MATCH1) source = ELLIPSE1;
    op_pk_send (pkptr, source);


    op_pk_send (queue[0][schedule_pt].pkptr, MEMORY);
    }

else if (queue[0][schedule_pt].status == Q_BUSY_RD)
    {
    pkptr = queue[0][schedule_pt].pkptr;
    op_pk_nfd_get (pkptr, "source", &source);
    op_pk_destroy (pkptr);
    if (source == MATCH0) source = ELLIPSE0;
    if (source == MATCH1) source = ELLIPSE1;
    op_pk_send (node[MEMORY].pkptr, source);
    }
```

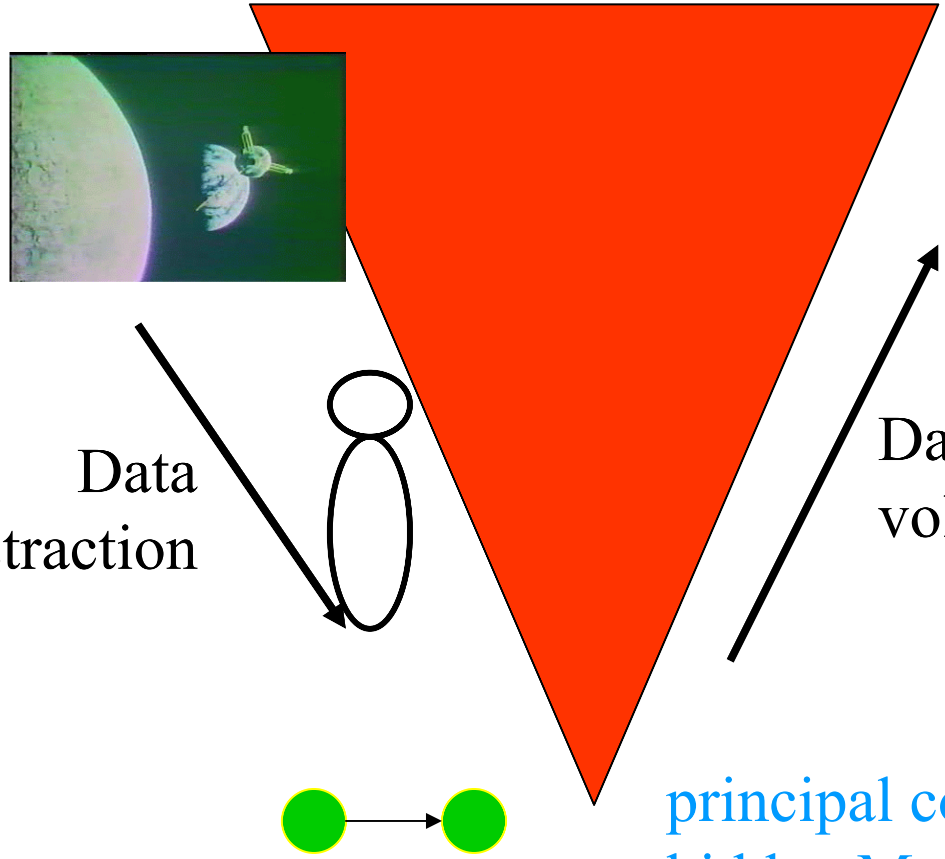
Power and area analysis

- Based on the network component library
  - Circuit model and layout for each component
- Power $P=\sum\sum N_{ij}*E_{ij}$
  - N_{ij} is the number of a type of activity of a network component (performance analysis)
  - E_{ij} is the energy consumed by a type of activity of a network component (circuit model)
- Metal area and silicon area $A=\sum M_j*S_j$
  - M_j is the number of a network component (floorplan estimation)
  - S_j is the area of a network component (layout)
- Most power are consumed by links
- Metal area is much larger than silicon area

But is it worth the effort?

- Performance/power is only one metric.
  - Must also consider design time, manufacturing volume.
- Design and mask costs push us toward fewer platforms.
- But those platforms have some common characteristics.

The multimedia processing funnel



pixel processing

Edge extraction

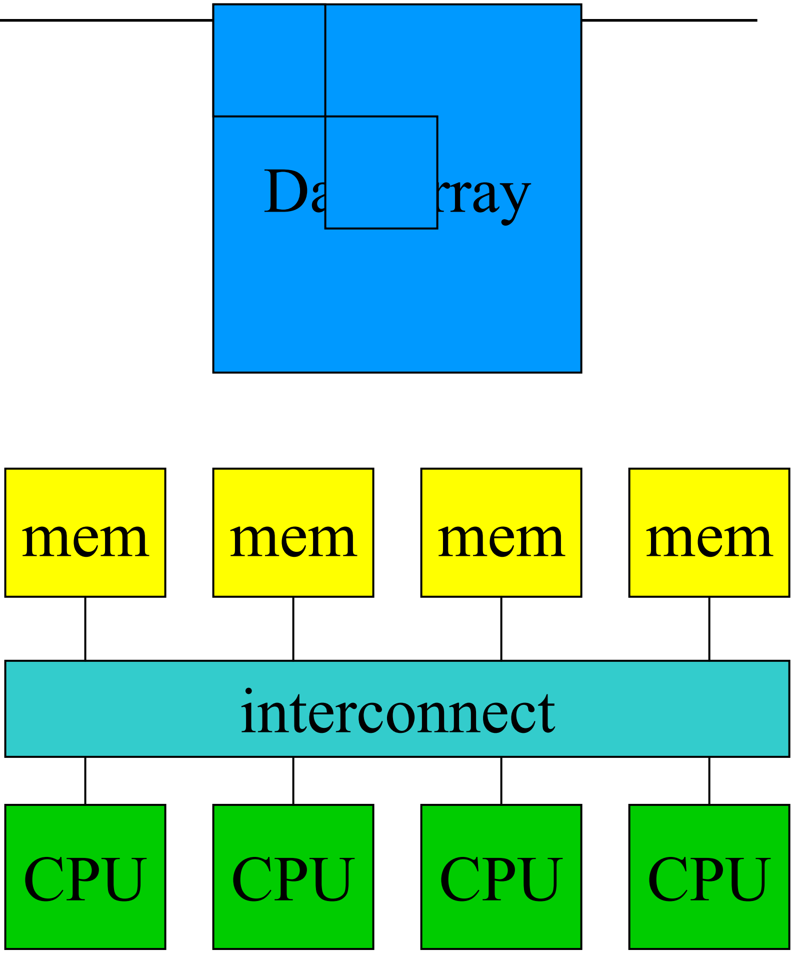
principal component analysis, hidden Markov models

Data abstraction

Data volume

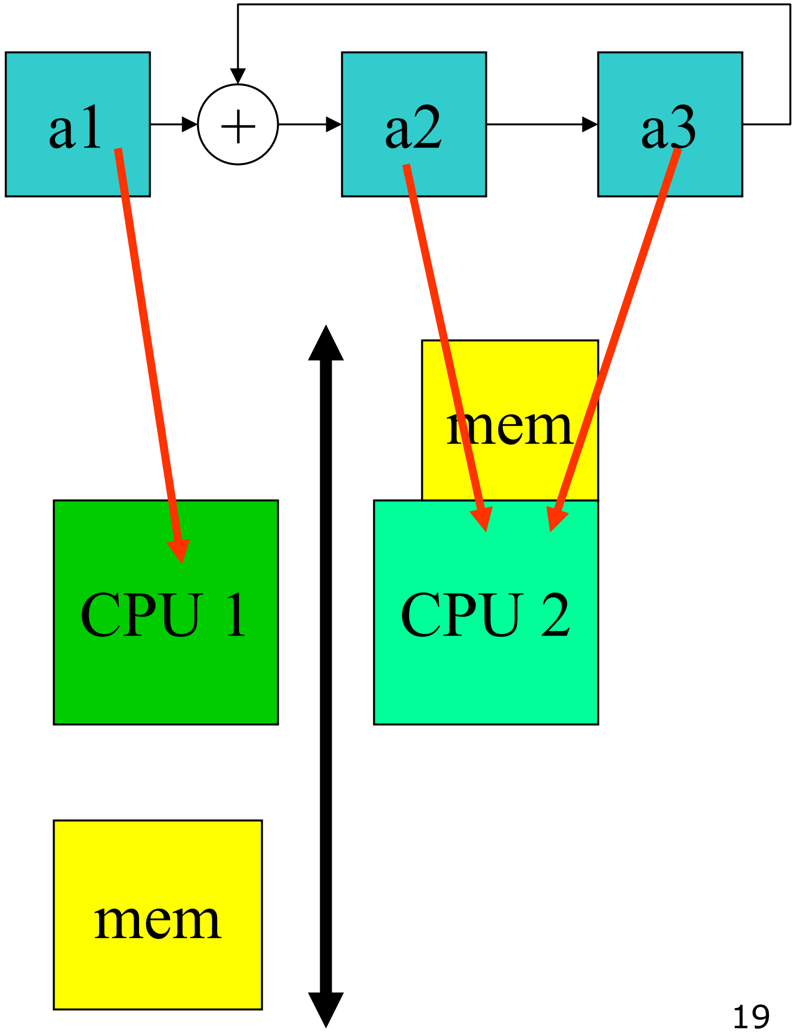
Scientific multiprocessing

- Traditional scientific algorithms perform numerical computations.
  - Single algorithm on large amounts of data.
- Scientific multiprocessors emphasize easy programming of a single data set over multiple CPUs.



Embedded vs. scientific applications

- Embedded applications provide task-level parallelism.
- Embedded applications run many different types of algorithms at once.
- Embedded applications need real-time and QoS.



Summary

- Application-specific networks can pay off in performance/energy.
  - Take more time to design, but may pay off.
- Methodology is very vertical:
  - Layout through architecture.