UNIVERSITY
OF OSLO

Exploring Knowledge Schemes for Efficient Evolution of Hardware

Jim Torresen
Department of Informatics
University of Oslo, Norway

2004

UNIVERSITY
OF OSLO

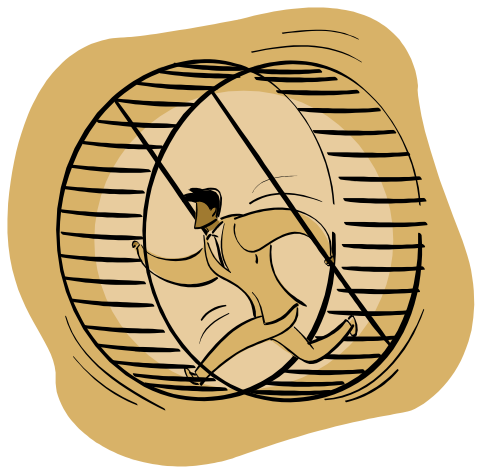
How do we solve a problem?

2004

UNIVERSITY
OF OSLO

How do we evolve a problem?

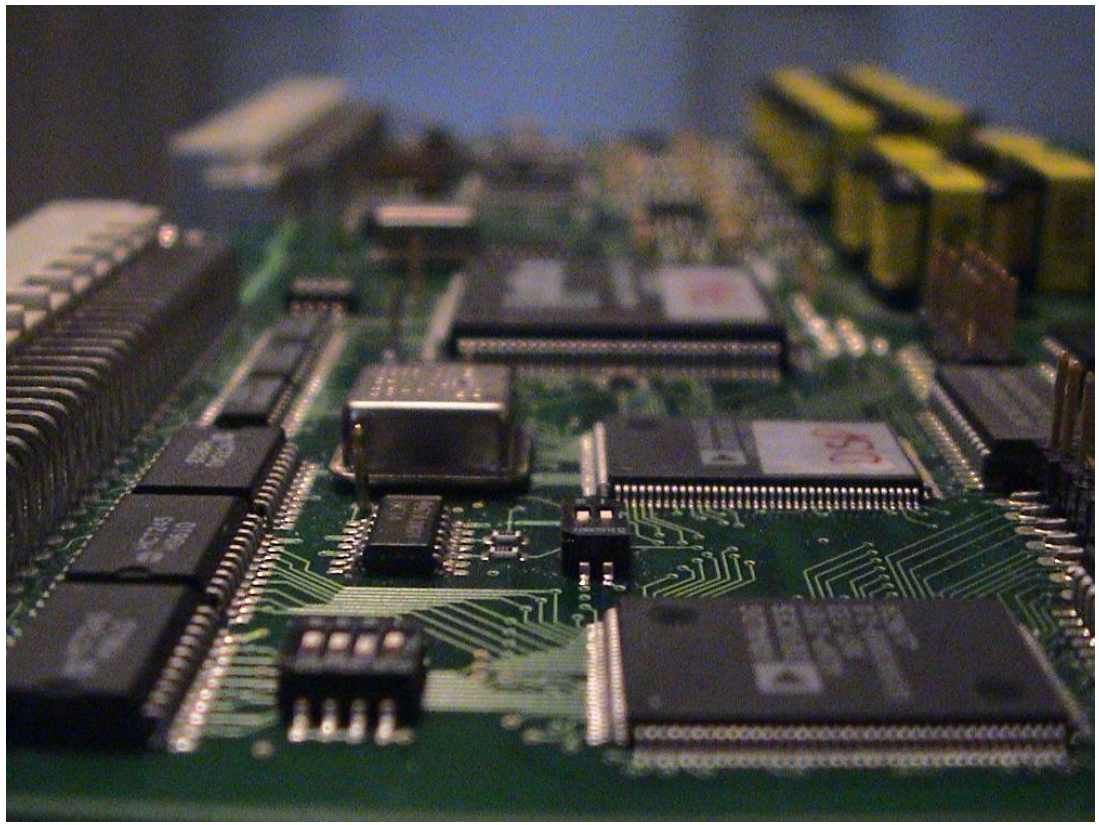
- Apply Evolutionary Algorithms with no knowledge about the problem!
- We avoid using available knowledge!
- In this talk, we will discuss how human knowledge can be added to evolution.



2004

UNIVERSITY
OF OSLO

What a priori knowledge is available?



2004

UNIVERSITY
OF OSLO

Design specification

2004

UNIVERSITY
OF OSLO

Partitioning the design task

2004

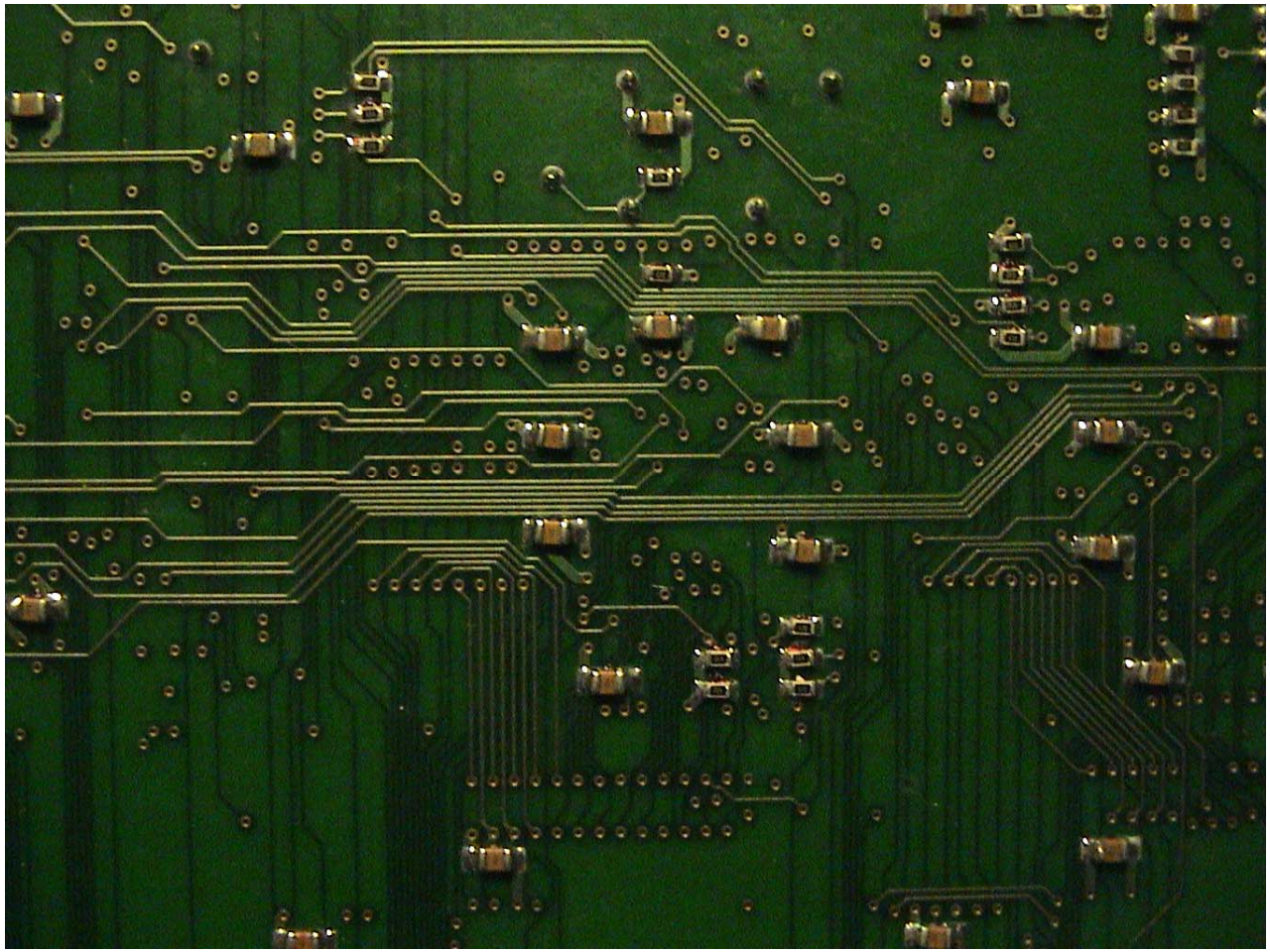
UNIVERSITY
OF OSLO

Reuse

2004

UNIVERSITY
OF OSLO

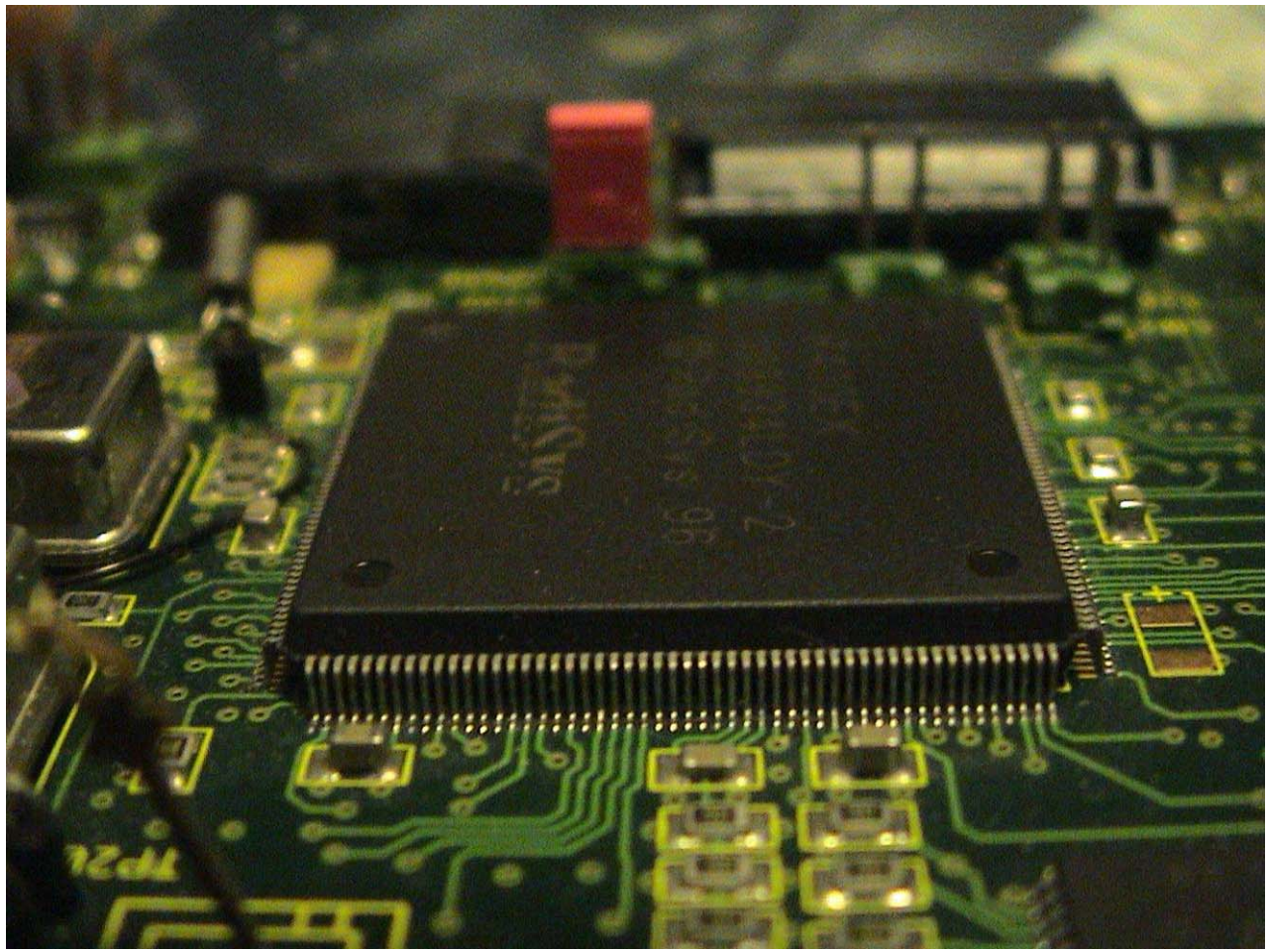
Data buses



2004

UNIVERSITY OF OSLO

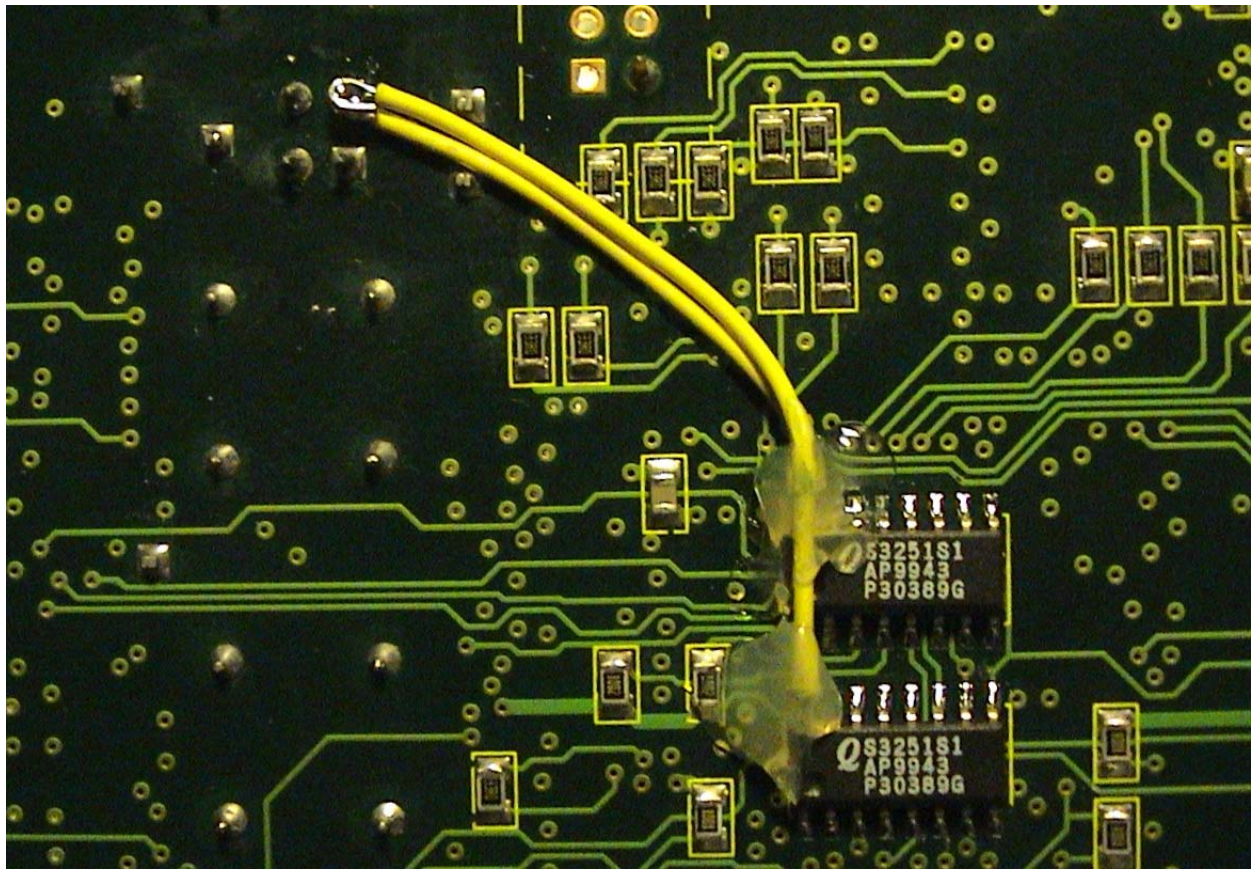
Design optimization knowledge



2004

UNIVERSITY OF OSLO

Prototyping



2004

UNIVERSITY OF OSLO

Hardware/software co-design

Jul 29, 03 15:15 **link.c** Page 3/4

```
      link.out.writeptr = link.out.buffer;
    //Increment the number of frames in line.
    link.out.waiting++;
    return(TRUE);
  }
  //The frame buffer is full.
  DELETE_FRAME(newframe);
  return(FALSE);
}
////////////////////////////////////////////////////////////////////////////////



////////////////////////////////////////////////////////////////////////////////
//Function:     Pop
//
//Parameters:   -buf:    The buffer to place the contents of the frame in
//              -length: The length of the buffer.
//
//Return Value: Integer value telling how many bytes that are read.
//
//Description:  Pop the first frame of the incomming frame buffer, strip
//              off the head and return the payload.
////////////////////////////////////////////////////////////////////////////////
uint8_t Pop(uchar_t* buf, uint8_t size){
  frame_t* frame;
  uint8_t  length;

  //If there are no frames in the incomming buffer,
  //then return zero.
  if(link.in.waiting == 0)
    return(0);
  //Else there are one ore more frames to process.
  //Pop of the first in line.
  frame = *link.in.readptr;
  //Get the length of the frame.
  length = frame->size;
  //Get the received frames data.
  if(length <= size)
    memcpy(buf, frame->body, size);
  else
    length = 0;
  //Decrease the number of frames in line.
  link.in.waiting--;
  //Increase or wrap the read pointer.
  link.in.readptr++;
  if(link.in.readptr == link.in.buffer+FRAMEBUF_SIZE)
    link.in.readptr = link.in.buffer;
  //Delete the frame, but not the body.
  DELETE_FRAME(frame);
  return(length);
}
////////////////////////////////////////////////////////////////////////////////
```

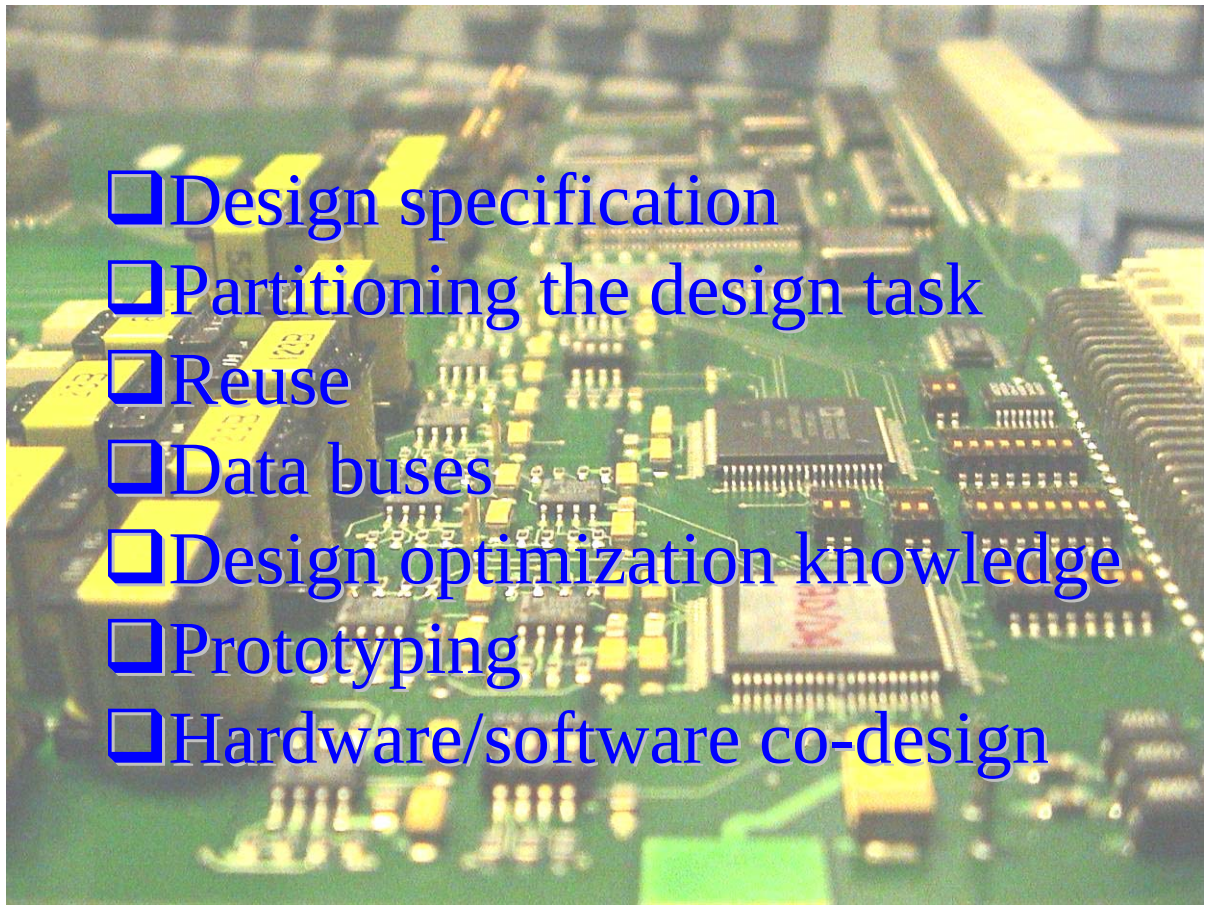
Jul 29, 03 15:06 **kernel.vhd** Page 13/24

```
        --The mode has been changed, return to home position
        --and change mode.
        rotation_Drive <= HALT;
        OpState        <= HOMING_START;
      end if;
------------------------------------------------------------------------------
when ASCAN_DIVERT =>
  --Set current status in status register.
  Status <= STAT_ASCAN;
  --Decide next state.
  if ElevationPosition = ElevationDest then
    --No change in elevation position.
    OpState <= ASCAN_CONTINUE;
  else
    --The desired elevation position has changed, reposition.
    rotation_Drive <= HALT;
    OpState        <= MERIDIAN_REPOS;
  end if;
------------------------------------------------------------------------------
when ASCAN_CONTINUE =>
  --Set current status in status register.
  Status <= STAT_ASCAN;
  --Start up rotation motor.
  rotation_Drive <= DRIVE;
  OpState        <= ASCAN_RUN;
------------------------------------------------------------------------------
------------------------------------------------------------------------------



------------------------------------------------------------------------------
-- Tracking mode.
------------------------------------------------------------------------------
when TRACK_INIT =>
  --Change value in the status register.
  Status <= STAT_TRACK;
  --Enable sampling registers.
  sampleClr <= '0';
  --Decide elevation direction.
  if ElevationDest > ElevationPosition then
    elevation_Direction <= FORWARD;
  else
    elevation_Direction <= BACKWARD;
  end if;
  --Decide rotation direction.
  if RotationDest < RotationPosition then
    rotation_Direction <= FORWARD;
  else
    rotation_Direction <= BACKWARD;
  end if;
  --Proceed to the next state.
  OpState <= TRACK_STEP;
------------------------------------------------------------------------------
```

2004

UNIVERSITY
OF OSLO

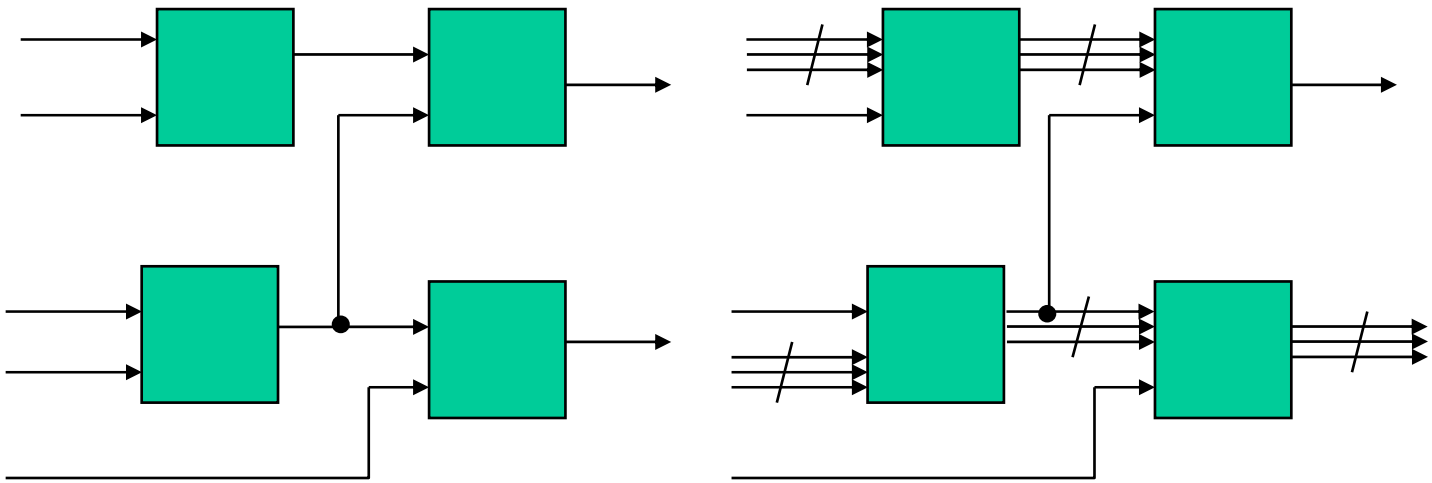
Complex design by a priori knowledge



- Design specification
- Partitioning the design task
- Reuse
- Data buses
- Design optimization knowledge
- Prototyping
- Hardware/software co-design

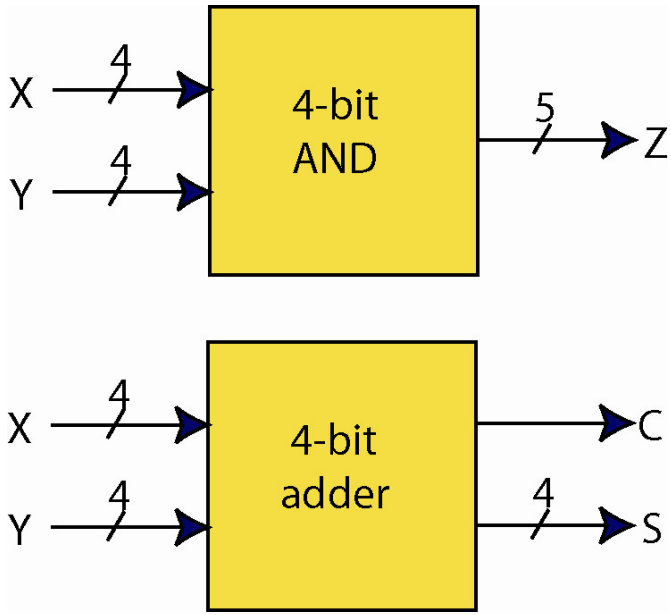
UNIVERSITY
OF OSLO

Evolution with Data Buses



UNIVERSITY OF OSLO

Building Blocks for Evolving a Multiplier



Inputs:

- 4 bits buses
- 1 bit lines

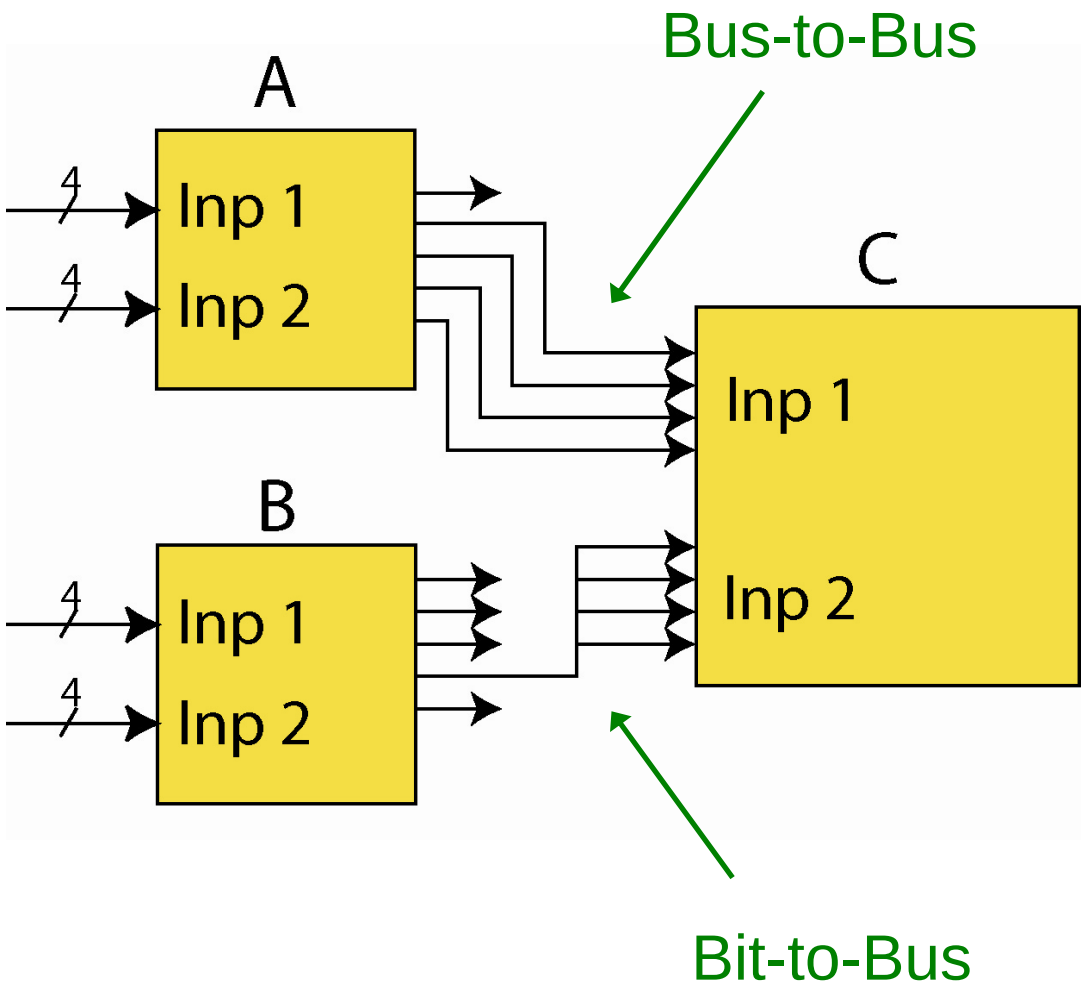
$$x_3x_2x_1x_0 \cdot y_3y_2y_1y_0$$

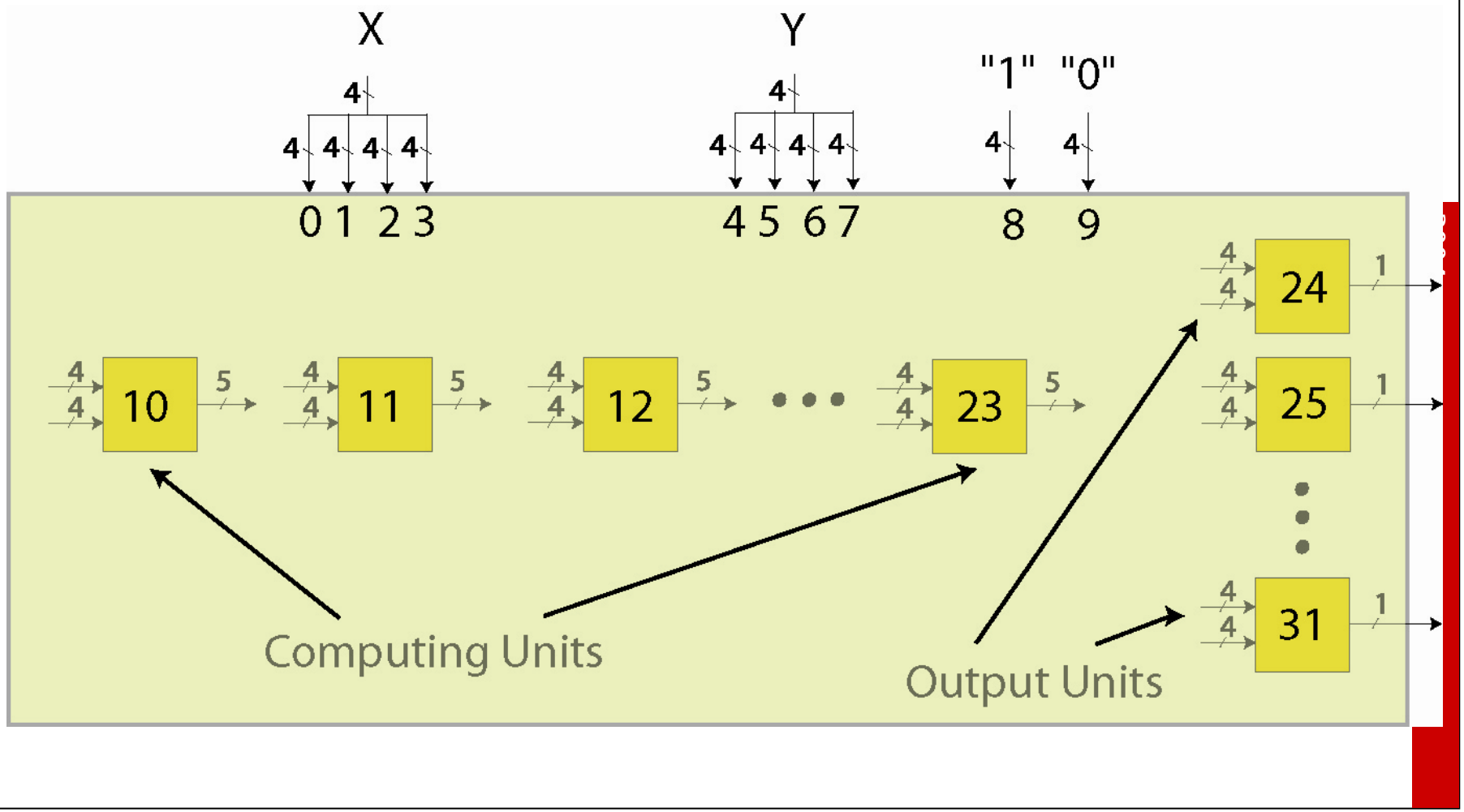
$$\begin{array}{cccccc} & & & x_0y_3 & x_0y_2 & x_0y_1 & x_0y_0 \\ & & x_1y_3 & x_1y_2 & x_1y_1 & x_1y_0 & \\ & x_2y_3 & x_2y_2 & x_2y_1 & x_2y_0 & & \\ + \; x_3y_3 & x_3y_2 & x_3y_1 & x_3y_0 & & & \\ \hline \end{array}$$

2004

UNIVERSITY
OF OSLO

Connection Between Units

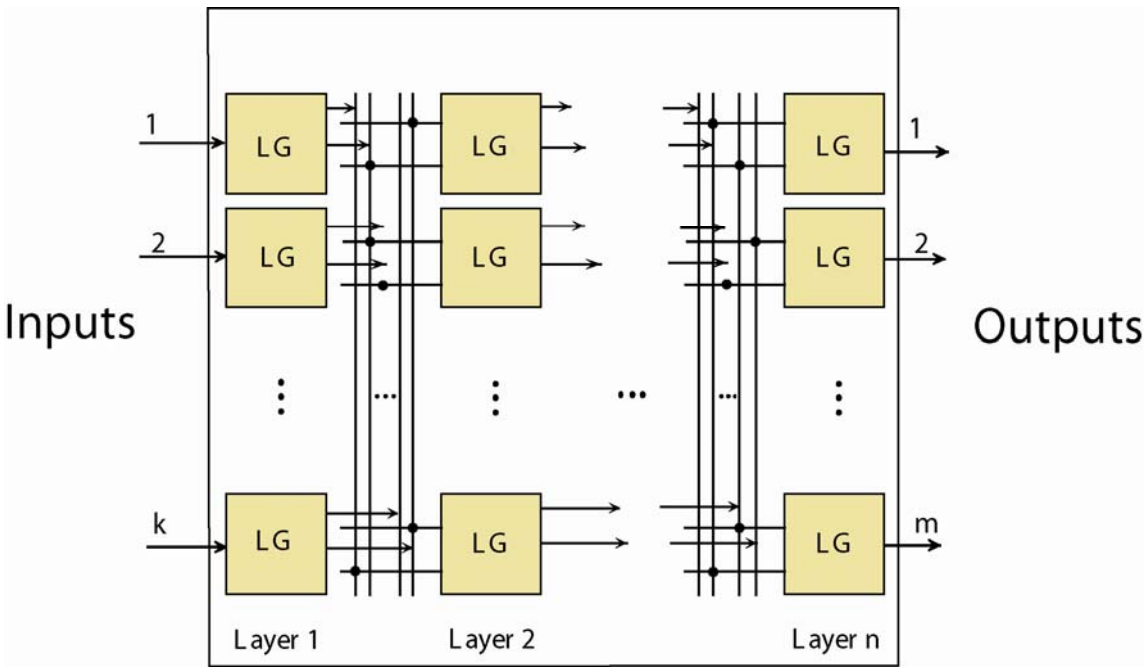



UNIVERSITY
OF OSLO
X
Y
"1"
"0"
4
4 4 4 4
0 1 2 3
4 5 6 7
8
9
10
11
12
23
24
25
31
5
1
Computing Units
Output Units

UNIVERSITY OF OSLO

Experiment

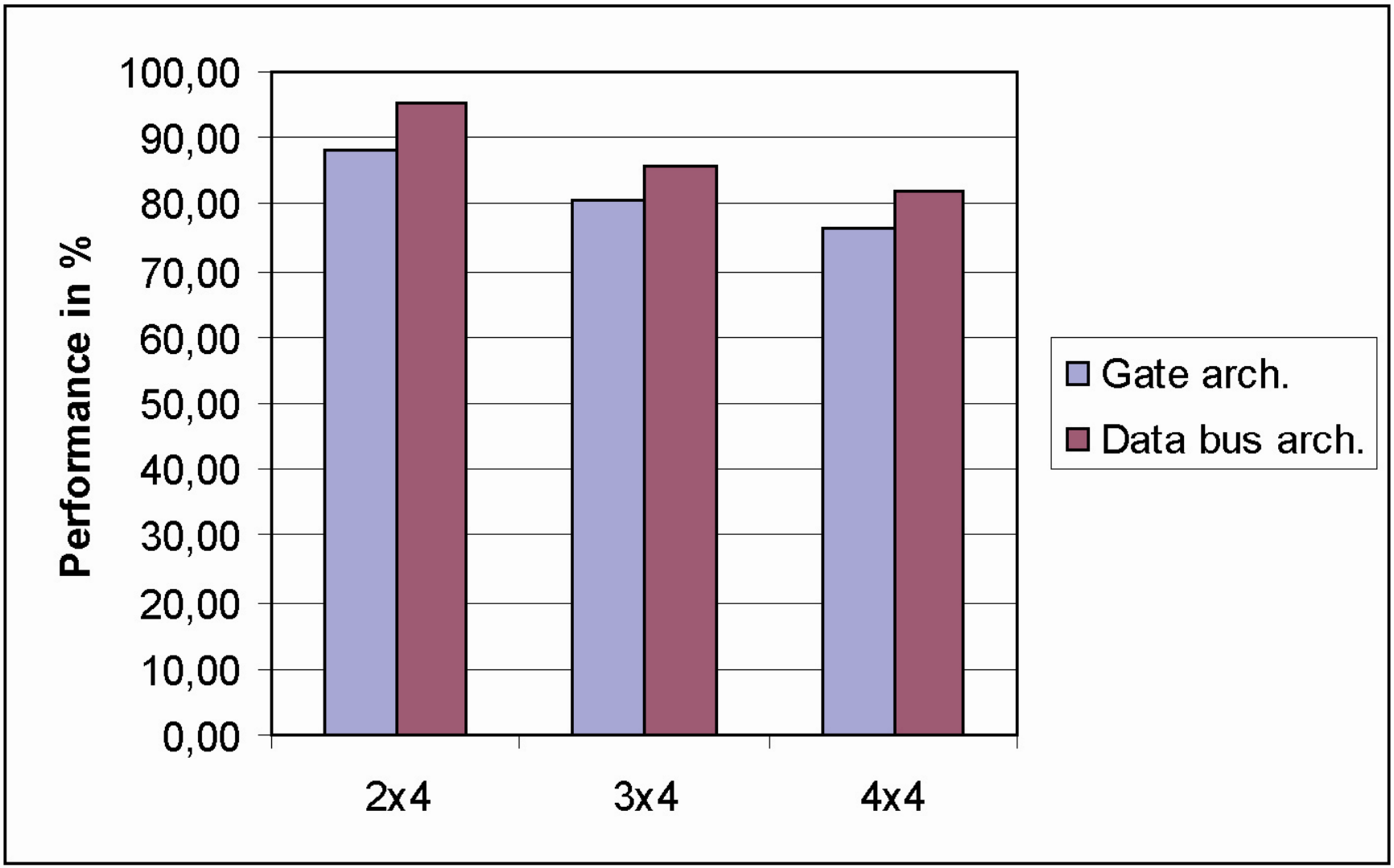
- Evolve multiplier circuits with genetic algorithms.
- Compare evolving the new architecture with gate array:



2004

UNIVERSITY
OF OSLO

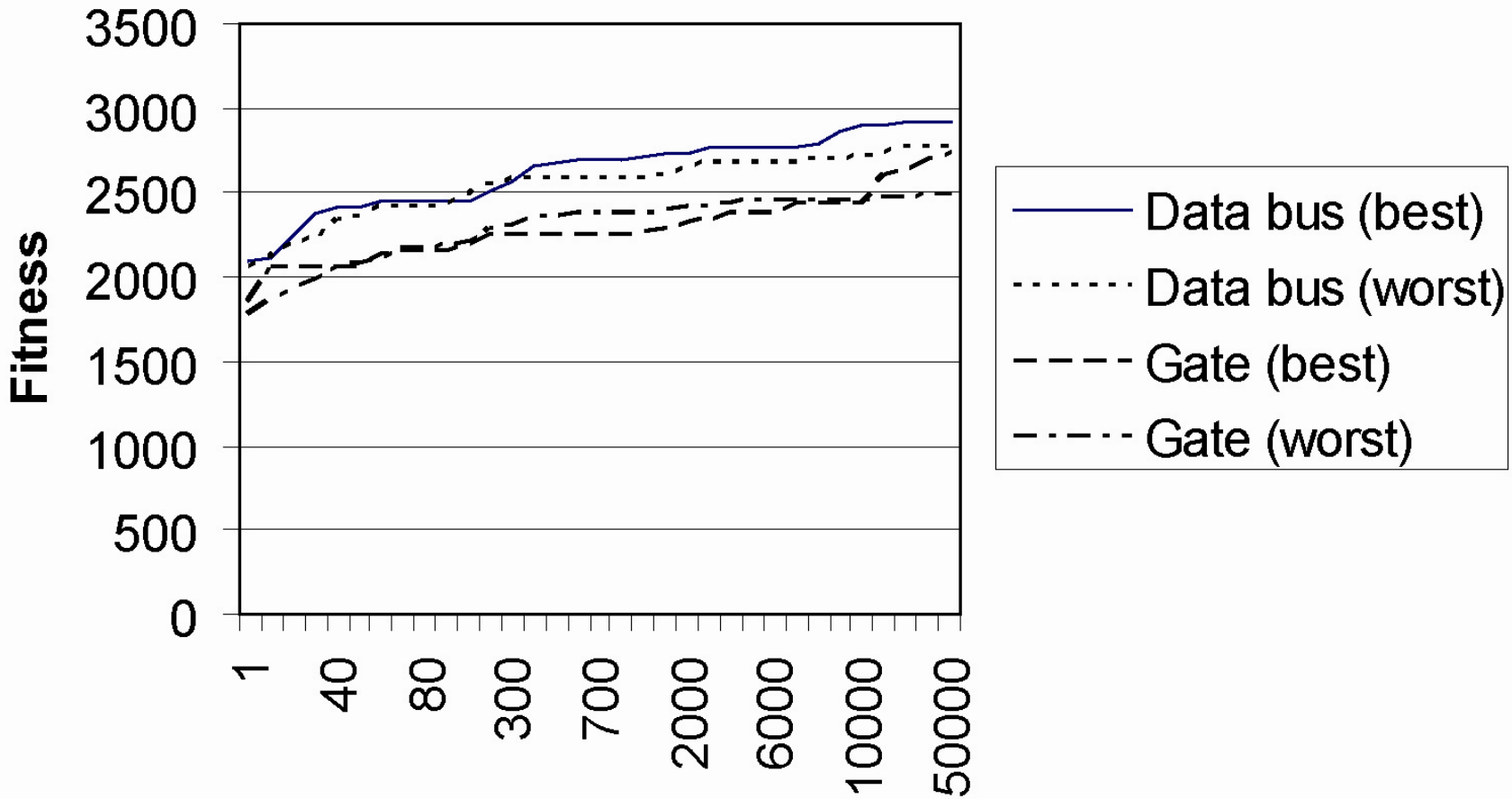
Results



2004

UNIVERSITY OF OSLO

Results: Best and Worst Circuit



UNIVERSITY OF OSLO

Conclusions

- A number of traditional design aspects have been discussed.
- Experiments involving *data buses* in evolution of multiplier circuits have been reported.
- Results are better than for traditional gate level evolution.
- Further development of the architecture and the a priory knowledge inclusion is future work.