Homework index
Goals for lecture

• Explain details of a real-time design problem
• Give some background on development of area
• Synthesis solution
• Current commercial status
Distributed real-time: Part one

- Distributed needn’t mean among cities or offices – Same IC?
- Process scaling trends
- Cross-layer design now necessary
Embedded system / SOC synthesis motivation

- Wireless: effects of the communication medium important
- Hard real-time: deadlines must not be violated
- Reliable: anti-lock brake controllers shouldn’t crash
- Rapidly implemented: IP use, simultaneous HW-SW development
- High-performance: massively parallel, using ASICs
- SOC market from $1.1 billion in 1996 to $14 billion in 2000 (Dataquest), to $43 billion in 2009 (Global Information, Inc.)
Global $\mu$-controller sales

Source: Embedded Processor and Microcontroller Primer and FAQ by Russ Hersch
Low-power motivation

• Embedded systems frequently battery-powered, portable
• High heat dissipation results in
  – Expensive, bulky packaging
  – Limited performance
• High-level trade-offs between
  – Power
  – Speed
  – Price
  – Area
Past embedded system synthesis work

- **Early 1990s**: Optimal MILP co-synthesis of small systems
  [Prakash & Parker], [Bender], [Schwiegershausen & Pirsch]

- **Mid 1990s**: One CPU-One ASIC
  [Ernst, Henkel & Benner], [Gupta & De Micheli]
  [Barros, Rosenstiel, & Xiong], [D’Ambrosio & Hu]

- **Late 1990s – present**: Co-synthesis of heterogeneous distributed embedded systems
  [Kuchcinski],
  [Quan, Hu, & Greenwood], [Wolf]
Past low-power work

• **Mid 1990s**: VLSI power minimization design surveys
  [Pedram], [Devadas & Malik]

• **Mid – late 1990s**: High-level power analysis and optimization
  [Raghunathan, Jha, & Dey], [Chandrakasan & Brodersen]

• **Late 1990s**: Embedded processor energy estimation
  [Li & Henkel], [Sinha & Chandrakasan]

• **Late 1990s – present**: Low-power hardware-software co-synthesis
  [Dave, Lakshminarayana, & Jha], [Kirrovski & Potkonjak]
Overview of system synthesis projects

- **TGFF**: Generates parametric task graphs and resource databases
- **MOGAC**: Multi-chip distributed systems
- **CORDS**: Dynamically reconfigurable
- **COWLS**: Multi-chip distributed, wireless, client-server
- **MOCSYN**: System-on-a-chip composed of hard cores, area optimized
Overview of system synthesis projects

• Synthesize embedded systems
  – heterogeneous processors and communication resources
  – multi-rate
  – hard real-time

• Optimize
  – price
  – power consumption
  – response time
Overview of system synthesis projects

- **TGFF**: Generates parametric task graphs and resource databases
- **MOGAC**: Multi-chip distributed systems
- **CORDS**: Dynamically reconfigurable
- **COWLS**: Multi-chip distributed, wireless, client-server
- **MOCSYN**: System-on-a-chip composed of hard cores, area optimized
Definitions

- Specify
  - task types
  - data dependencies
  - hard and soft task deadlines
  - periods
- Analyze performance of each task on each resource
- Allocate resources
- Assign each task to a resource
- Schedule the tasks on each resource
Definitions

- Specify
  - task types
  - data dependencies
  - hard and soft task deadlines
  - periods

- Analyze performance of each task on each resource
- Allocate resources
- Assign each task to a resource
- Schedule the tasks on each resource
Allocation

Number and types of:
- PEs or cores
- Commun. resources
Assignment

- Assignment of tasks to PEs
- Connection of communication resources to PEs
Assignment

• Assignment of tasks to PEs
• Connection of communication resources to PEs
k, l, and n need not be scheduled
Costs

Soft constraints:
- price
- power
- area
- response time

Hard constraints:
- deadline violations
- PE overload
- unschedulable tasks
- unschedulable transmissions

Solutions which violate hard constraints not shown to designer – pruned out.
Genetic algorithms

- Multiple solutions
- Local randomized changes to solutions
- Solutions share information with each other
- Can escape sub-optimal local minima
- Scalable
Cluster genetic operator constraints motivation

Solution A

Solution B

PE type

Solution A

Solution B

PE type

PE allocation

Task assignment
Cluster genetic operator constraints motivation

Solution A

Solution B

PE type

DCT
DIV
FIR

Cut

DCT
DIV
FIR

PE allocation

Task assignment
Cluster genetic operator constraints motivation

Solution A

Solution B

PE type

Task allocation

Cut

PE type

Task assignment
Cluster genetic operator constraints

- Task assignment crossover
- PE allocation mutation
- Communication resource allocation mutation
- Communication resource connectivity mutation
- Communication resource connectivity crossover
- PE allocation crossover
- Task assignment mutation
- Solution
- Cluster
Locality in solution representation

A, B, and C attributes each solve sub-problems
Locality in solution representation

Cut

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
</tr>
</thead>
</table>
| Soln. 1

Cut

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
</tr>
</thead>
</table>
| Soln. 2

Cut

<table>
<thead>
<tr>
<th>A1</th>
<th>B1</th>
<th>C1</th>
<th>A2</th>
<th>B2</th>
<th>C2</th>
<th>A3</th>
<th>B3</th>
<th>C3</th>
</tr>
</thead>
</table>
| Soln. 1

Cut

<table>
<thead>
<tr>
<th>A1</th>
<th>B1</th>
<th>C1</th>
<th>A2</th>
<th>B2</th>
<th>C2</th>
<th>A3</th>
<th>B3</th>
<th>C3</th>
</tr>
</thead>
</table>
| Soln. 2
Information trading

- PE type
- Power consumption
- Price

Random orientation

90°

Swap

Don’t swap

27
A solution dominates another if all its costs are lower, i.e.,
\[ \text{dom}_{a,b} = \forall_{i=1}^{n} \text{cost}_{a,i} < \text{cost}_{b,i} \land a \neq b \]

A solution’s rank is the number of other solutions which do not dominate it, i.e.,
\[ \text{rank}_{s'} = \sum_{i=1}^{n} \neg \text{dom}_{s_i,s'} \]
Multiobjective optimization

Linear cost functions
\[ \sum_{i=1}^{n} w_i \cdot \text{cost}_i \]

Non-linear cost functions
\[ \max_{i=1}^{n} w_i \cdot \text{cost}_i \]

Pareto-rank cost function
\[ \sum_{i=1}^{n} \text{not dom}_{s_i, s'} \]
Solution are selected for reproduction by conducting Boltzmann trials between parents and children.

Given a global temperature $T$, a solution with rank $J$ beats a solution with rank $K$ with probability:

$$\frac{1}{1 + e^{(K-J)/T}}$$
MOCSYN related work

- Floorplanning block placement – Fiduccia and Mattheyses, 1982
  – Stockmeyer, 1983
- Parallel recombinative simulated annealing – Mahfoud and Goldberg, 1995
- Linear interpolating clock synthesizers – Bazes, Ashuri, and Knoll, 1996
- Interconnect performance estimation models – Cong & Pan, 2001
MOCSYN algorithm overview

Clock selection

Initialization

Change core allocation

Results

Cluster loop

Task prioritization

Communication assignment

Schedule

Change task assignment

Architecture loop

Link re-prioritization

Bus structure

Block placement

Link prioritization
MOCSYN algorithm overview

Cluster loop
- Link re-prioritization
- Bus structure
- Block placement
- Link prioritization

Architecture loop
- Change task assignment
- Schedule
- Communication assignment
- Task prioritization

Initialization
- Clock selection
- Change core allocation
- Results

Change task assignment
- Link prioritization

Schedule
- Cluster loop

Communication assignment
- Architecture loop

Task prioritization
- Change task assignment

Clock selection
- Initialization
- Change core allocation
- Results
Clock selection

- Cores have different maximum frequencies
- Globally synchronous system forces underclocking
- Multiple crystals too expensive
- Use linear interpolating clock synthesizers
  - Standard CMOS process
  - Each core runs near highest speed
  - Global clock frequency can be low to reduce power
- Optimal clock selection algorithm in pre-pass
MOCSYN algorithm overview

Cluster loop

- Clock selection
- Initialization
- Change core allocation

Architecture loop

- Task prioritization
- Communication assignment
- Schedule
- Change task assignment

Link
- Link re-prioritization
- Bus structure
- Block placement
- Link prioritization

Results
MOCSYN algorithm overview

Cluster loop

- Clock selection
- Initialization
- Change core allocation
- Results

Architecture loop

- Task prioritization
- Communication assignment
- Schedule
- Change task assignment
- Change core allocation
- Link prioritization
- Link re-prioritization
- Bus structure
- Block placement
MOCSYN algorithm overview

Cluster loop

Clock selection
Initialization
Change core allocation
Results

Architecture loop

Task prioritization
Communication assignment
Schedule
Change task assignment

Link re-prioritization
Bus structure
Block placement
Link prioritization
Link prioritization

Estimate communication time based on average core separation.

Duration

Quantity

Deadline = 20 ms

Slack = 2 ms
Priority = -2
MOCSYN algorithm overview

Block placement to determine communication time, energy
Floorplanning block placement

Balanced binary tree of cores formed
Division takes into account:

- Link priorities
- Area of cores on each side of division
Floorplanning block placement
Floorplanning block placement
MOCSYN algorithm overview

Bus topology generation: minimize contention under routability constraints
Bus formation

Use efficient red-black tree data structure for intersection tests
RMST bus length reduction

Total length = 5.6 mm

Merge

Total length = 2.1 mm
Bus formation

Highest density

Link pri = 7

Link pri = 5

Merge

Link pri = 12
MOCSYN algorithm overview
Task prioritization

Deadline = 20 ms

Slack = 3 ms
Priority = −3
Scheduling

- Fast list scheduler
- Multi-rate
- Handles period $< \text{deadline}$ as well as period $\geq \text{deadline}$
- Uses alternative prioritization methods: slack, EST, LFT
- Other features depend on target

Time

3 copies

Period = 20 ms
Deadline = 20 ms

Period = 30 ms
Deadline = 40 ms

2 copies

System hyperperiod = 60 ms
Cost calculation

- Price
- Average power consumption
- Area
- PE overload
- Hard deadline violation
- Soft deadline violation
- etc.
Clock selection quality

Average proportion of maximum internal frequencies

External frequency (MHz)

8X frequency multiplication
No frequency multiplication
### MOCSYN feature comparisons experiments

<table>
<thead>
<tr>
<th>Example</th>
<th>MOCSYN price ($)</th>
<th>Worst-case commun. price ($)</th>
<th>Best-case commun. price ($)</th>
<th>Single bus price ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>15</td>
<td>216</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td>16</td>
<td>138</td>
<td>n.a.</td>
<td>n.a.</td>
<td>177</td>
</tr>
<tr>
<td>17</td>
<td>283</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td>18</td>
<td>253</td>
<td>n.a.</td>
<td>n.a.</td>
<td>253</td>
</tr>
<tr>
<td>19</td>
<td>211</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Better</td>
<td></td>
<td>38</td>
<td>44</td>
<td>28</td>
</tr>
<tr>
<td>Worse</td>
<td></td>
<td>3</td>
<td>1</td>
<td>9</td>
</tr>
</tbody>
</table>

17 processors, 34 core types, five task graphs, 10 tasks each, 21 task types from networking and telecomm examples.
### MOCSYN multiobjective experiments

<table>
<thead>
<tr>
<th>Example</th>
<th>Price ($)</th>
<th>Average power (mW)</th>
<th>Soft DL viol. prop.</th>
<th>Area (mm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>automotive-industrial</td>
<td>91</td>
<td>120</td>
<td>0.60</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td>91</td>
<td>120</td>
<td>0.61</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>110</td>
<td>113</td>
<td>0.88</td>
<td>4.0</td>
</tr>
<tr>
<td></td>
<td>110</td>
<td>115</td>
<td>0.60</td>
<td>4.0</td>
</tr>
<tr>
<td>networking</td>
<td>61</td>
<td>72</td>
<td>0.94</td>
<td>38.4</td>
</tr>
<tr>
<td>telecomm</td>
<td>223</td>
<td>246</td>
<td>2.31</td>
<td>9.9</td>
</tr>
<tr>
<td></td>
<td>223</td>
<td>246</td>
<td>2.76</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>233</td>
<td>255</td>
<td>3.47</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>236</td>
<td>247</td>
<td>2.29</td>
<td>9.9</td>
</tr>
<tr>
<td></td>
<td>236</td>
<td>249</td>
<td>2.60</td>
<td>8.0</td>
</tr>
<tr>
<td></td>
<td>242</td>
<td>221</td>
<td>2.67</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td>242</td>
<td>230</td>
<td>2.44</td>
<td>25.9</td>
</tr>
<tr>
<td></td>
<td>242</td>
<td>237</td>
<td>1.72</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>272</td>
<td>226</td>
<td>2.22</td>
<td>192.1</td>
</tr>
<tr>
<td></td>
<td>272</td>
<td>226</td>
<td>2.34</td>
<td>9.4</td>
</tr>
<tr>
<td></td>
<td>353</td>
<td>258</td>
<td>1.23</td>
<td>4.0</td>
</tr>
<tr>
<td>consumer</td>
<td>134</td>
<td>281</td>
<td>1.40</td>
<td>34.1</td>
</tr>
<tr>
<td></td>
<td>134</td>
<td>281</td>
<td>1.50</td>
<td>21.6</td>
</tr>
<tr>
<td>office automation</td>
<td>64</td>
<td>370</td>
<td>0.23</td>
<td>36.8</td>
</tr>
<tr>
<td></td>
<td>66</td>
<td>55</td>
<td>0.00</td>
<td>7.2</td>
</tr>
</tbody>
</table>
MOGAC run on Hou’s examples

<table>
<thead>
<tr>
<th>Example</th>
<th>Yen’s System</th>
<th>MOGAC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Price ($)</td>
<td>CPU Time (s)</td>
</tr>
<tr>
<td>Hou 1 &amp; 2 (unclustered)</td>
<td>170</td>
<td>10,205</td>
</tr>
<tr>
<td>Hou 3 &amp; 4 (unclustered)</td>
<td>210</td>
<td>11,550</td>
</tr>
<tr>
<td>Hou 1 &amp; 2 (clustered)</td>
<td>170</td>
<td>16.0</td>
</tr>
<tr>
<td>Hou 3 &amp; 4 (clustered)</td>
<td>170</td>
<td>3.3</td>
</tr>
</tbody>
</table>

Robust to increase in problem complexity.

2 task graphs each example, 3 PE types
Unclustered: 10 tasks per task graph  Clustered: approx. 4 tasks per task graph
### MOGAC run on Prakash & Parker’s examples

<table>
<thead>
<tr>
<th>Example 〈Perform〉</th>
<th>Prakash &amp; Parker’s System</th>
<th>MOGAC</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Price ($)</td>
<td>CPU Time (s)</td>
<td>Price ($)</td>
</tr>
<tr>
<td>Prakash &amp; Parker 1 〈4〉</td>
<td>7</td>
<td>28</td>
<td>7</td>
</tr>
<tr>
<td>Prakash &amp; Parker 1 〈7〉</td>
<td>5</td>
<td>37</td>
<td>5</td>
</tr>
<tr>
<td>Prakash &amp; Parker 2 〈8〉</td>
<td>7</td>
<td>4,511</td>
<td>7</td>
</tr>
<tr>
<td>Prakash &amp; Parker 2 〈15〉</td>
<td>5</td>
<td>385,012</td>
<td>5</td>
</tr>
</tbody>
</table>

Quickly gets optimal when getting optimal is tractable.

3 PE types, Example 1 has 4 tasks, Example 2 has 9 tasks
# MOGAC run Yen’s large random examples

<table>
<thead>
<tr>
<th>Example</th>
<th>Yen’s System</th>
<th>MOGAC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Price ($)</td>
<td>CPU Time (s)</td>
</tr>
<tr>
<td>Random 1</td>
<td>281</td>
<td>10,252</td>
</tr>
<tr>
<td>Random 2</td>
<td>637</td>
<td>21,979</td>
</tr>
</tbody>
</table>

Handles large problem specifications.

No communication links: communication costs = 0

Random 1: 6 task graphs, approx. 20 tasks each, 8 PE types
Random 2: 8 task graphs, approx. 20 tasks each, 12 PE types
MOCSYN contributions, conclusions

First core-based system-on-chip synthesis algorithm

• Novel problem formulation
• Multiobjective (price, power, area, response time, etc.)
• New clocking solution
• New bus topology generation algorithm

Important for system-on-chip synthesis to do

• Clock selection
• Block placement
• Generalized bus topology generation
Research contributions

• **TGFF**: Used by a number of researchers in published work

• **MOGAC**: Real-time distributed embedded system synthesis
  – First true multiobjective (price, power, etc.) system synthesis
  – Solution quality \( \geq \) past work, often in orders of magnitude less time

• **CORDS**: First reconfigurable systems synthesis, schedule reordering

• **COWLS**: First wireless client-server systems synthesis, task migration
EEMBC-based embedded benchmarks

Automotive-Industrial

Processors

- AMD ElanSC520 133 MHz
- AMD K6-2 450 MHz
- AMD K6-2E 400MHz/ACR
- AMD K6-2E+ 500MHz/ACR
- AMD K6-IIIE+ 550MHz/ACR
- Analog Devices 21065L 60 MHz
- IBM PowerPC 405GP 266 MHz
- IBM PowerPC 750CX 500 MHz
- IDT32334 100 MHz
- IDT79RC32364 100 MHz
- IDT79RC32V334 150 MHz
- IDT79RC64575 250 MHz
- Imsys Cjip 40 MHz
- Motorola MPC555 40 MHz
- NEC VR5432 167 MHz
- ST20C2 50 MHz
- TI TMS320C6203 300MHz
Recently started and future work

• Market-based energy allocation in low-power wireless mobile networks
  – paper under review

• Evolutionary algorithms for multi-dimensional optimization
  – future work

• Task and processor characterization
  – EEMBC-based resource database completed will publicly release

• Tightly coupling low-level, high-level design automation algorithms
  – recently started work in this area
MOGAC run on Yen’s second large random example
MOCSYN Networking example

Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
MOCSYN Networking example

Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
MOCSYN Networking example

Price, power, and area only. Soft deadline violation omitted.
MOCSYN Networking example

Price, power, and area only. Soft deadline violation omitted.
MOCSYN Networking example

Price, power, and area only. Soft deadline violation omitted.
MOCSYN Networking example

Price, power, and area only. Soft deadline violation omitted.
MOCSYN Networking example

Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
MOCSYN Networking example

Price, power, and area only. Soft deadline violation omitted.
MOCSYN Networking example

Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
MOCSYN Networking example

Price, power, and area only. Soft deadline violation omitted.
MOCSYN Networking example

Price, power, and area only. Soft deadline violation omitted.
MOCSYN Networking example

Price, power, and area only. Soft deadline violation omitted.
MOCSYN Networking example

Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
MOCSYN Networking example

Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
MOCSYN Networking example

Price, power, and area only. Soft deadline violation omitted.
MOCSYN Networking example

Price, power, and area only. Soft deadline violation omitted.
Price, power, and area only. Soft deadline violation omitted.
Problem complexity

Allocations:
\[ \text{max\_}PE\_\text{per\_}type^{\text{max\_}PE\_\text{types}} \cdot \text{max\_link\_per\_type}^{\text{max\_link\_types}} \]

Assignments:
\[ \Theta \left( \text{PE\_count}^{\text{task\_count}} \right) \]

Link Connectivities:

- Consider each PE to be a node in a graph
- Each link is a group which can contain up to \( \text{max\_contacts\_per\_link} \) nodes

\[ \Theta \left( C(\text{PE\_count}, \text{max\_contacts\_per\_link})^{\text{link\_count}} \right) \]
Take a simple system:

\[
\begin{align*}
\text{max PE per type} &= \text{max link per type} = 3 \\
\text{max PE types} &= \text{max link types} = 3 \\
\text{PE count} &= \text{link count} = 9 \\
\text{task count} &= 10 \\
\text{max contacts per link} &= 2 \\
\end{align*}
\]

\[
\begin{align*}
\text{allocations} &= 3^3 \cdot 3^3 = 27 & \text{good} \\
\text{assignments} &= \Theta(9^{10}) = \Theta(3.49 \times 10^9) & \text{bad} \\
\text{connectivities} &= \Theta(C(9, 2)^9) = \Theta(1.02 \times 10^{14}) & \text{worse}
\end{align*}
\]

Number of architectures to evaluate:

\[
\Theta\left(27 \cdot 3.49 \times 10^9 \cdot 1.02 \times 10^{14}\right) = \Theta\left(9.57 \times 10^{24}\right)
\]

\ldots and this does not even take scheduling complexity or multi-core ICs into account
Counter-division only clock selection

Max Freq. 80 MHz
Actual Freq. 50 MHz

Reference = 50 MHz
Quality = 0.707

Reference = 80 MHz
Quality = 0.867
Counter-division only clock selection

Reference = 100 MHz
Quality = 0.875

Reference = 150 MHz
Quality = 0.896
Bus formation inner kernel

$l$ is number of communicating core pairs

For each bus, $i$, intersecting with highest density point: $\mathcal{O}(l^2)$

For each bus, $j$: $\mathcal{O}(l^3)$
  - Tentatively merge $i$ and $j$: $\mathcal{O}(l^4)$
  - Evaluate the density, $new\_dens$, of congest: $\mathcal{O}(l^3)$
  - Evaluate new maximum contention estimate, $cont\_est$: $\mathcal{O}(l^4)$

If $new\_dens$ decreased for any tentative merge:
  - Merge the pair with greatest $new\_dens$ decrease: $\mathcal{O}(l^2)$
  - Break ties by selecting merge with least $cont\_est$ increase.