PARALLELIZATION STRATEGIES FOR WIRELESS BASEBAND

MARTIN PALKOVIC

ARTISTDESIGN MPSOC CLUSTER MEETING

7TH JULY 2010
WIRELESS DOMAIN CONTAINS “SEA” OF STANDARDS; AND CUSTOMERS WANT TO HAVE THEM ALL IN ONE DEVICE
FOR MULTI-MODE >100MBPS WIRELESS COMMUNICATION SOFTWARE DEFINED RADIO (SDR) SOLUTIONS ARE CRUCIAL

Software Defined Radio

- DSP
- ASIP
- ASIC

Flexibility vs. Energy Efficiency
THERE IS A LONG WAY TILL STANDARD SPECIFICATION IS MAPPED ON THE SDR PLATFORM
OUTLINE

IMEC SDR BEAR platform
Baseband SW mapping flow
- Sequential MATLAB code
- Sequential C code
- Parallel C code
IMEC SDR COBRA platform
Conclusions
OUR PLATFORM DEALS WITH THE STREAM FROM THE ANTENNA RECEIVER TILL THE FORWARD ERROR CORRECTION ENGINE

DFE = Digital Front-End
ARM = ARM processor
FEC = Forward Error Correction

SPM = Scratch-Pad Memory
L2 = Layer 2 memory
ADRES = ADRES processor
TO OBTAIN ENERGY EFFICIENCY ON THE PLATFORM, COMBINATION OF DSP, ASIP AND ASIC IS USED

Energy Efficiency

Flexibility

- Modulation
  - Demodulation
  - ADRES, ARM proc.

- Synchronization
  - Signal detection
  - DFE ASIP

- Forward Error Correction
  - FEC ASIC
THE BASEBAND PART OF OUR SDR PLATFORM, WE WILL FOCUS ON, FEATURES ALL DIFFERENT LEVELS OF PARALLELISM
OUTLINE

IMEC SDR BEAR platform
Baseband SW mapping flow
- Sequential MATLAB code
- Sequential C code
- Parallel C code
IMEC SDR COBRA platform
Conclusions
OUTLINE

IMEC SDR BEAR platform
Baseband SW mapping flow
- Sequential MATLAB code
- Sequential C code
- Parallel C code
IMEC SDR COBRA platform
Conclusions
KERNEL SELECTION FROM THE CODE GENERATED BY MATLAB2C CONVERSION TOOL

WLAN.c
fft.c
chan_comp.c
demap.c
DATA LEVEL PARALLELISM (DLP) AND INSTRUCTION LEVEL PARALLELISM (ILP) ARE EXPLOITED IN THE KERNEL CODE

DLP
▶ Intrinsics

ILP
▶ DRESC Compiler
OUR MODULO SCHEDULING ALGORITHM IN THE DRESC COMPILER IS THE KEY TO EFFICIENT ILP EXPLOITATION

Loop body:
for (i=0;i<n;i++)
z = (x+y)*a + (x+y)/b;

Where to place an operation? (placement)
When to schedule an operation? (scheduling)
How to connect operations? (routing)

Modulo-Scheduling (Space-time representation)

Initiation Interval (II) = 1
Pipeline stages = 3
4 operations/cycle for kernel
Optimized kernels are used in the skeleton code which was obtained by MATLAB2C conversion tool.

Optimized kernel library

Skeleton code

fftO.DRE

chan_compO.DRE
demapO.DRE

WLAN.c
OUTLINE

IMEC SDR BEAR platform
Baseband SW mapping flow
- Sequential MATLAB code
- Sequential C code
- Parallel C code
IMEC SDR COBRA platform
Conclusions
STILL, WLAN 40 MHZ MIMO NOT REAL-TIME FOR 1 ADRES. TASK LEVEL PARALLELISM (TLP) IS NOT EXPLOITED YET!

Input stream:

<table>
<thead>
<tr>
<th>Preamble</th>
<th>Symb1</th>
<th>Symb2</th>
<th>Symb3</th>
<th>Symb4</th>
<th>…</th>
</tr>
</thead>
<tbody>
<tr>
<td>12µs</td>
<td>4µs</td>
<td>4µs</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Decoding time:

5.8µs

ARM

TLP
THE APPLICATION OFFERS MULTIPLE WAYS FOR PARALLELIZATION. WHICH ONE TO CHOOSE?

Ant1

Ant2

Preamble Symb1 Symb2 Symb3 Symb4 ...

Preamble Symb1 Symb2 Symb3 Symb4 ...

time&freq offset
↓
FFT
↓
chan.est.
↓
inv.

time&freq offset
↓
FFT
↓
tracking
↓
chan.comp.
↓
demap.

→ to next symbol

→ to FEC
ANALYZING THE COMMUNICATION FOR POTENTIAL SPLITS IS ALWAYS A GOOD THING TO DO!

Per antenna split 100MB/s (1/3 bus)

Odd-even symbol split 2.5MB/s (<1%)
CREATION OF THE PARALLEL CODE AND FAST EVALUATION ARE HARD THINGS FOR THE DESIGNER – TOOL SUPPORT NEEDED!
HIGH-LEVEL SIMULATOR PROVIDES FAST INFORMATION ABOUT THE QUALITY OF OUR PARALLELIZATION

<table>
<thead>
<tr>
<th>Thread activity</th>
<th>Thread activity</th>
<th>Thread activity</th>
<th>Thread activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1_preamble</td>
<td>T2_symbol_odd</td>
<td>T3_symbol_even</td>
<td>thread_walking_info</td>
</tr>
<tr>
<td>W1_preamble</td>
<td>W2_symbol_odd</td>
<td>W3_symbol_even</td>
<td>thread_walking_info</td>
</tr>
<tr>
<td>W1_preamble</td>
<td>W2_symbol_odd</td>
<td>W3_symbol_even</td>
<td>thread_walking_info</td>
</tr>
<tr>
<td>K1_preamble</td>
<td>K2_symbol_odd</td>
<td>K3_symbol_even</td>
<td>thread_walking_info</td>
</tr>
<tr>
<td>FPO usage</td>
<td>FPO usage</td>
<td>FPO usage</td>
<td>FPO usage</td>
</tr>
<tr>
<td>P0_15_T3_cram 31:0</td>
<td>P0_15_T3_cram 31:0</td>
<td>P0_15_T3_cram 31:0</td>
<td>P0_15_T3_cram 31:0</td>
</tr>
<tr>
<td>P0_15_T3_cram 31:0</td>
<td>P0_15_T3_cram 31:0</td>
<td>P0_15_T3_cram 31:0</td>
<td>P0_15_T3_cram 31:0</td>
</tr>
<tr>
<td>P0_15_T3_cram 31:0</td>
<td>P0_15_T3_cram 31:0</td>
<td>P0_15_T3_cram 31:0</td>
<td>P0_15_T3_cram 31:0</td>
</tr>
<tr>
<td>P0_15_T3_cram 31:0</td>
<td>P0_15_T3_cram 31:0</td>
<td>P0_15_T3_cram 31:0</td>
<td>P0_15_T3_cram 31:0</td>
</tr>
</tbody>
</table>

arp.data

HLSim lib

./par/* .c
WE IMPLEMENTED THE FIFO COMMUNICATION VIA THE VLIW AND HALT INTERRUPTS OF ADRES ENGINES

Preamble
Odd symbols
Even symbols

ADRES1
1
3
...

ADRES2
2
4
...

VLIW Int ADR.1
ARM
Release ADR.2
(if halted)
Copy data

Halt int. ADR.2
VLIW int ADR.1
VLIW int ADR.2

Halt int ADR.1
VLIW int ADR.1, Halt int ADR.2

Data SPM
ADRES1

Data SPM
ADRES2

1
2
3
4
OUTLINE

IMEC SDR BEAR platform
Baseband SW mapping flow
- Sequential MATLAB code
- Sequential C code
- Parallel C code

IMEC SDR COBRA platform
Conclusions
COBRA PLATFORM LEARNED FROM THE DRAWBACKS OF BEAR
CONCLUSIONS

Multi-mode >100Mbps wireless communication requires multi-processor software-defined radio (SDR) solutions

Parallelization for SDR should be exploited on each level – SIMD (DLP), ILP and TLP

Odd-even symbol group split seems to be the right choice for TLP for 40 MHz MIMO SDM OFDM

ILP was explored by our robust C DRESC compiler and TLP split was supported by our MPA parallelization tool

With combination of DLP, ILP and TLP we achieved real-throughput behavior of 40 MHz MIMO SDM OFDM

Cobra platform is the natural evolution of Bear to >500Mbps, multistream and run-time reconfiguration