SCA Based Implementation of STANAG 4285 in a Joint Effort under the NATO RTO/IST Panel

Sarvpreet Singh, Marc Adrat, Stefan Couturier, Markus Antweiler

FGAN/FKIE, Germany
singh@fgan.de

Martin Phisel, Steve Bernier

CRC, Canada
phisel@crc.ca

SDR’08, Technical Conference and Product Exposition
Washington, D.C., October 26th – 30th, 2008
Outline

- NATO Group
- Motivation
- Tools and Waveform
- SCA Implementations (Different Granularity)
- Profiling Results
- Conclusion
NATO RTO/IST-080 RTG-038 on SDR

- NATO RTO/IST RTG on SDR (estd. 2007)
  - Research and Technology Organisation (RTO)
  - Informations Systems Technology (IST)
  - Regular Task Group (RTG)

- Main Objectives
  - Share knowledge and experience on SCA/SDR Development
  - Share waveforms and SCA based waveform components
  - Report results to others groups like NC3B SDRUG or SDR Forum
Motivation

- Target Workflow

1. SCA-based implementation of STANAG 4285 waveform

2. Demonstrate portability onto national SDR platforms

3. Demonstrate interoperability between the different implementations
Motivation

- Target Workflow

1. SCA-based implementation of STANAG 4285 waveform

2. Demonstrate portability onto national SDR platforms

3. Demonstrate interoperability between the different implementations
STANAG 4285 – A NATO HF Waveform

- Offers six modes between 75 bits/sec and 2.4 kbits/sec

Diagram:

Frame Collection:
- AGC
- SOM
- User Data
- EOM
- Flush Bits

Frame Processing:
- FEC
- Interleaving
- Modulation

Configuration Parameters:
- Code Rate
- Interleaver Rate
- PSK Scheme

Switch Encoder State
SCA Architect (CRC, Canada)

- An IDE as a plug-in to Eclipse Framework
- SCA Compliant Component Development
- Create Graphical Models of various Elements
- Source Code Generation
- Assemble elements into applications and nodes
- Focus more on self functionality and not on SCA
Profiling Tool – Valgrind (Open Source)

- Valgrind – Framework for building dynamic analysis tools
  - Detect Memory Management Bugs
  - Detect Threading Bugs
  - Program Profiling

- Callgrind – Cache Profiler
  - Cache misses
  - Memory References
  - Instructions Executed

- KCachegrind – Visualizer
Profiling Tool – Valgrind (Open Source)

- Valgrind – Framework for building dynamic analysis tools
  - Detect Memory Management Bugs
  - Detect Threading Bugs
  - Program Profiling

- Callgrind – Cache Profiler
  - Cache misses
  - Memory References
  - Instructions Executed

- KCachegrind – Visualizer
SCA based Implementation of STANAG 4285

- SCA based implementations of different TX granularity level

Diagram:

- "Cool, the quick ...
- STANAG 4285 Tx
- PSK, Coding, Interleaving
- STANAG 4285 Rx
- PSK, Coding, Interleaving
- I/Q-Values or IF-signal

SCA-based on PC
SCA based Implementation of STANAG 4285

- SCA based implementations of different TX granularity level

Diagram:

SCA-based on PC

PSK, Coding, Interleaving → STANAG 4285 Tx → I/Q-Values or IF-signal → STANAG 4285 Rx → PSK, Coding, Interleaving

"Cool, the quick ...

Collection → Processing → STANAG 4285 Rx → PSK, Coding, Interleaving

"Cool, the quick ...

Proceeding of the SDR 08 Technical Conference and Product Exposition. Copyright © 2008 SDR Forum. All Rights Reserved
SCA based Implementation of STANAG 4285

- SCA based implementations of different TX granularity level

SCA-based on PC

```
PSK, Coding, Interleaving
↓
STANAG 4285 Tx

PSK, Coding, Interleaving
↓
STANAG 4285 Rx

"Cool, the quick ...

I/Q-Values or IF-signal
```

SCA-based on PC

```
PSK, Coding, Interleaving
↓
Collection

Processing

PSK, Coding, Interleaving
↓
STANAG 4285 Rx

"Cool, the quick ...

I/Q-Values or IF-signal
```

SCA-based on PC

```
PSK, Coding, Interleaving
↓
Collection

FEC

Interleaving

Modulation

PSK, Coding, Interleaving
↓
STANAG 4285 Rx

"Cool, the quick ...

I/Q-Values or IF-signal
```
Profiling Results for 1 Component TX

Frame Collection
- AGC
- SOM
- User Data
- FOM
- Flush Bits

Frame Processing
- FEC
- Interleaving
- Modulation

Configuration Parameters:
- Code Rate
- Interleaver Rate
- PSK Scheme

Graph:
- Exe
- TAO
- ACE
- C/C++
- SCA
- ld.so

Self Costs vs. Number of Frames
Profiling Results for 1 Component TX

Frame Collection
- AGC
- SOM
- User Data
- FOM
- Flush Bits

Frame Processing
- Encoder
- Switch
- Configuration Parameters:
  - Code Rate
  - Interleaver Rate
  - PSK Scheme

- FEC
- Interleaving
- Modulation

Legend:
- Exe
- TAO
- ACE
- C/C++
- SCA
- ld.so

Graph:
- Number of Frames vs. Self Costs
- Exe: ★
- TAO: ◇
- ACE: △
- C/C++: ○
- SCA: ▽
- ld.so: □

Graph Range:
- 0 to 5 × 10^4 Self Costs
- 1 × 10^4 to 2 × 10^4 Number of Frames
Profilng Results for 1 Component TX

Number of Frames: 2

Self Costs:
- Exe: $1 \times 10^4$
- TAO: $2 \times 10^4$
- ACE: $3 \times 10^4$
- C/C++: $4 \times 10^4$
- SCA: $5 \times 10^4$

Number of Frames vs. Overhead:
- Exe
- TAO
- ACE
- C/C++
- SCA

Overhead vs. Number of Frames:
- Exe
- TAO
- ACE
- C/C++
- SCA

Configuration Parameters:
- Code Rate
- Interleaver Rate
- PSK Scheme
Profiling Results for 1 Component TX

![Diagram showing profiling results with markers for Exe, TAO, ACE, C/C++, SCA, and ld.so.]
Profiling Results for 2 Component TX

![Diagram showing frame collection and processing with cost data.

- Frame Collection:
  - AGC
  - SOM
  - User Data
  - FOM
  - Flush Bits

- Frame Processing:
  - FEC
  - Interleaving
  - Modulation

Configuration Parameters:
- Code Rate
- Interleaver Rate
- PSK Scheme

Graph showing costs for different numbers of frames:
- Exe
- TAO
- ACE
- C/C++
- SCA
- Id.so

Number of Frames:
- 1
- 2

Self Costs:
- 1
- 2
- 3
- 4
- 5
Profilling Results for 2 Component TX

1. Frame Collection
   - AGC
   - SOM
   - User Data
   - FOM
   - Flush Bits

2. Frame Processing
   - FEC
   - Interleaving
   - Modulation

Configuration Parameters:
- Code Rate
- Interleaver Rate
- PSK Scheme

**Graphs**
- **Self Costs**
  - X-axis: Number of Frames
  - Y-axis: Self Costs
- **Legend**
  - Exe
  - TAO
  - ACE
  - C/C++
  - SCA
  - ld.so

CRC

RESEARCH INSTITUTE FOR COMMUNICATION, INFORMATION PROCESSING, AND ERGONOMICS
Communication Systems

Proceeding of the SDR 08 Technical Conference and Product Exposition. Copyright © 2008 SDR Forum. All Rights Reserved
Profiling Results for 2 Component TX

Frame Collection
- AGC
- SOM
- User Data
- FOM
- Flush Bits

Frame Processing
- HEQ
- Interleaving
- Modulation

Configuration Parameters:
- Code Rate
- Interleaver Rate
- PSK Scheme

Exe  TAO
ACE  C/C++
SCA  ld.so
Profiling Results for 4 Component TX

Frame Collection
- AGC
- SOM
- User Data
- FOM
- Flush Bits

Frame Processing
- Switch
- Encode
- State
- Configuration Parameters: Code Rate, Interleaver Rate, PSK Scheme
- Interleaving
- Modulation
- FEC

Graphs 1, 2, 3, 4:
- Number of Frames vs. Self Costs
- Logarithmic scale
- Data points for Exe, TAO, ACE, C/C++, SCA, ld.so

Legend:
- Exe
- TAO
- ACE
- C/C++
- SCA
- ld.so

Graph 1:
- Exe
- TAO
- ACE
- C/C++
- SCA
- ld.so

Graph 2:
- Exe
- TAO
- ACE
- C/C++
- SCA
- ld.so

Graph 3:
- Exe
- TAO
- ACE
- C/C++
- SCA
- ld.so

Graph 4:
- Exe
- TAO
- ACE
- C/C++
- SCA
- ld.so

Number of Frames:
- $10^4$
- $2 	imes 10^4$

Self Costs:
- $10^4$
- $2 	imes 10^4$
- $3 	imes 10^4$
- $4 	imes 10^4$
- $5 	imes 10^4$
Comparison of Profiling Results

![Graph showing self costs and relative overhead vs. number of frames.]

\[ r = \frac{\text{Overhead Self Cost}}{\text{Exec. + Ov. Self Cost}} \]
Comparison of Profiling Results

![Graph showing comparison of profiling results for 1TX and 2TX configurations. The x-axis represents the number of frames, and the y-axis represents self costs and relative overhead (r). The formula for relative overhead (r) is given as r = Overhead Self Cost / Exec. + Ov. Self Cost.](attachment:image.png)
Comparison of Profiling Results

\[ r = \frac{\text{Overhead Self Cost}}{\text{Exec. + Ov. Self Cost}} \]

![Graph showing the comparison of self costs and relative overhead for 1TX, 2TX, and 4TX across different numbers of frames.](image-url)
Conclusions

- SCA based implementation of STANAG 4285 as a NATO effort
  - Three implementations of different granularity level available

- Profiling results
  - Division creates Overheads
  - Self costs of both, overhead and executable, increase linearly with the amount of processed data
  - Self costs can contain Fixed Costs and Variable Costs
  - Fixed Costs are a burden if less user data is processed

- Recommendations
  - Perform considerable Signal Processing in each SCA resource
  - Avoid large number of resource divisions
Questions?

Many Thanks for
Your Attention ...

Thanks to
- Telefunken Racoms for providing the C code for STANAG 4285
- all the RTO group members for their valuable comments