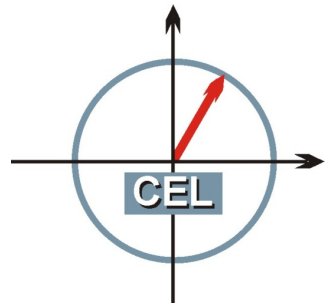
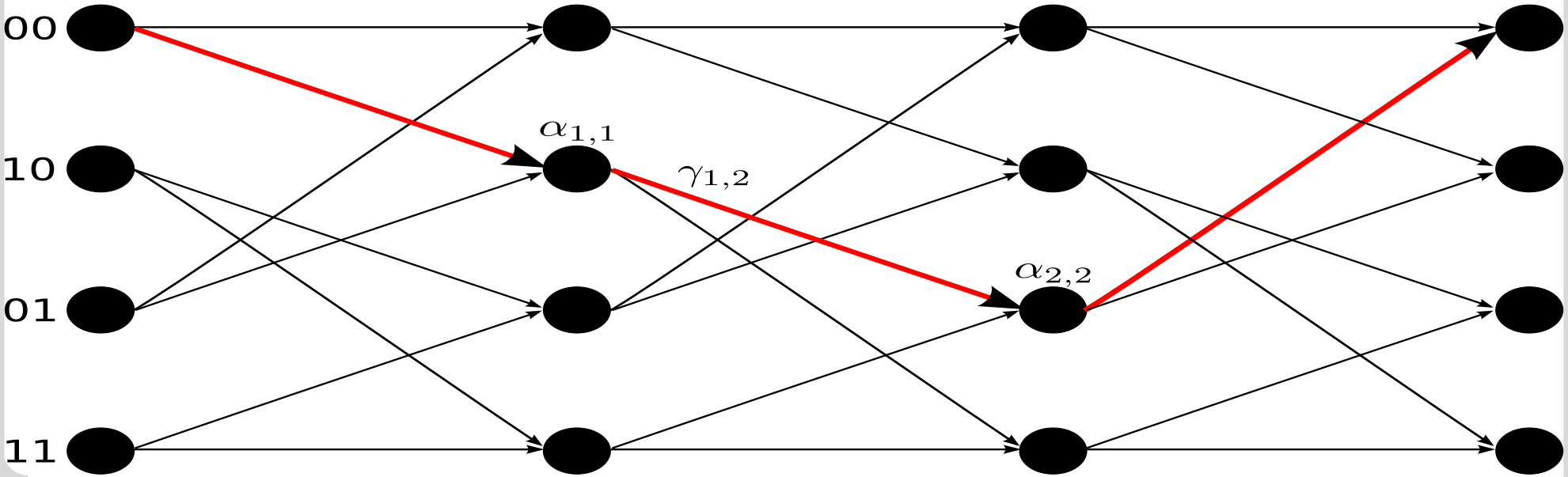
Karlsruhe Institute of Technology



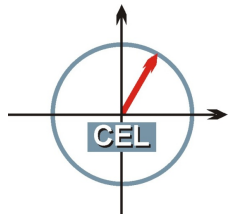
A Generic Yet Fast SIMD Viterbi-Decoder Or My Try To Square The Circle

Jan Kraemer | September 11, 2014

COMMUNICATIONS ENGINEERING LAB



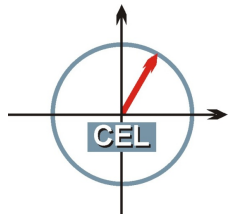
Why speed up the decoding process



- Viterbi Decoding is a very complex process
  - Neccessary for robust and fast communication
  - Bottleneck regarding throughput
- Gr-Trellis
  - Generic
  - Not optimized

$\Rightarrow$ More throughput would be nice!

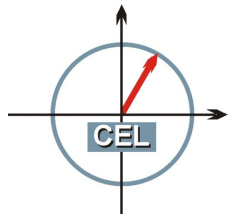
Why speed up the decoding process



- Viterbi Decoding is a very complex process
  - Neccessary for robust and fast communication
  - Bottleneck regarding throughput
- Gr-Trellis
  - Generic
  - Not optimized

$\Rightarrow$ More throughput would be nice!

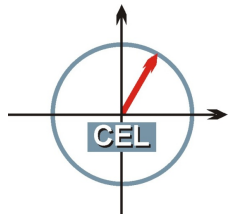
Why speed up the decoding process



- Viterbi Decoding is a very complex process
  - Neccessary for robust and fast communication
  - Bottleneck regarding throughput
- Gr-Trellis
  - Generic
  - Not optimized

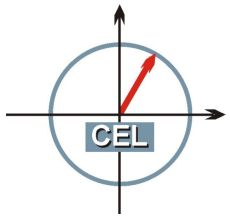
$\Rightarrow$ More throughput would be nice!

Gr-Trellis



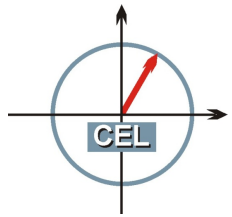
- Works with literally every trellis based encoding scheme
- Uses several lookup tables to be as generic as possible
- Already implemented in GNURadio
- So let's just modify it!

Gr-Trellis



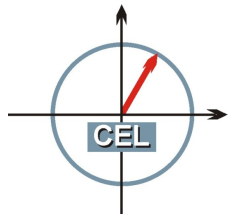
- Works with literally every trellis based encoding scheme
- Uses several lookup tables to be as generic as possible
- Already implemented in GNURadio
- So let's just modify it!

Gr-Trellis



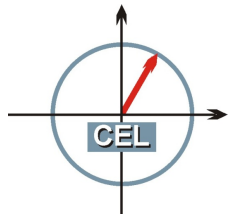
- Works with literally every trellis based encoding scheme
- Uses several lookup tables to be as generic as possible
- Already implemented in GNURadio
- So let's just modify it!

Gr-Trellis



- Works with literally every trellis based encoding scheme
- Uses several lookup tables to be as generic as possible
- Already implemented in GNURadio
- So let's just modify it!

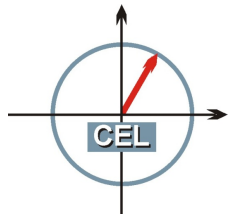
Amazing Results?



NO!

But Why?

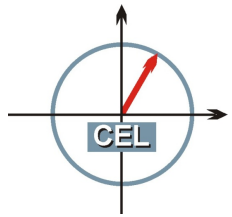
Amazing Results?



NO!

But Why?

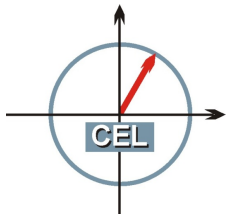
I blame the Lookup table



$$alpha[alphai * S + PS[j][i]] + in[k * O + OS[PS[j][i] * I + PI[j][i]]]$$

- SSE Operations work better with coherent memory
- Generic lookup tables do not fulfill this requirement

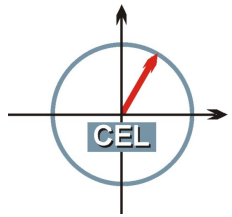
Start from scratch



Assumptions

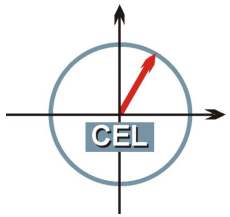
- Rate 1/n codes
- Non-recursive codes
- Number of states $\geqslant 8$

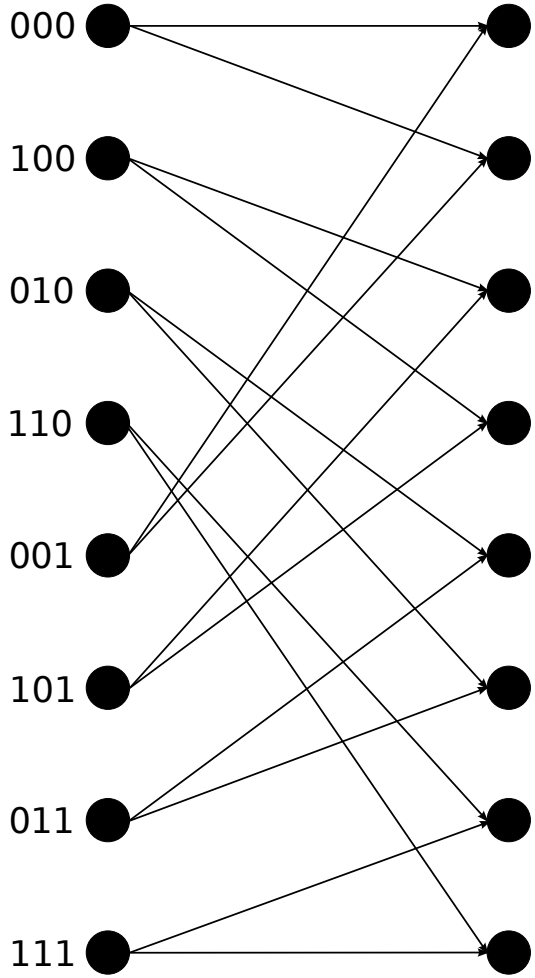
Episode 4: A new decoder



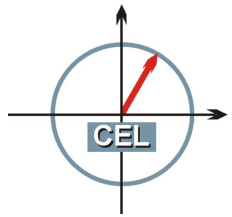
- Integration into the new FEC-API
- Based on Phillip Karns and FEC-APIs Voyager Decoder
- Uses a SSE float implementation

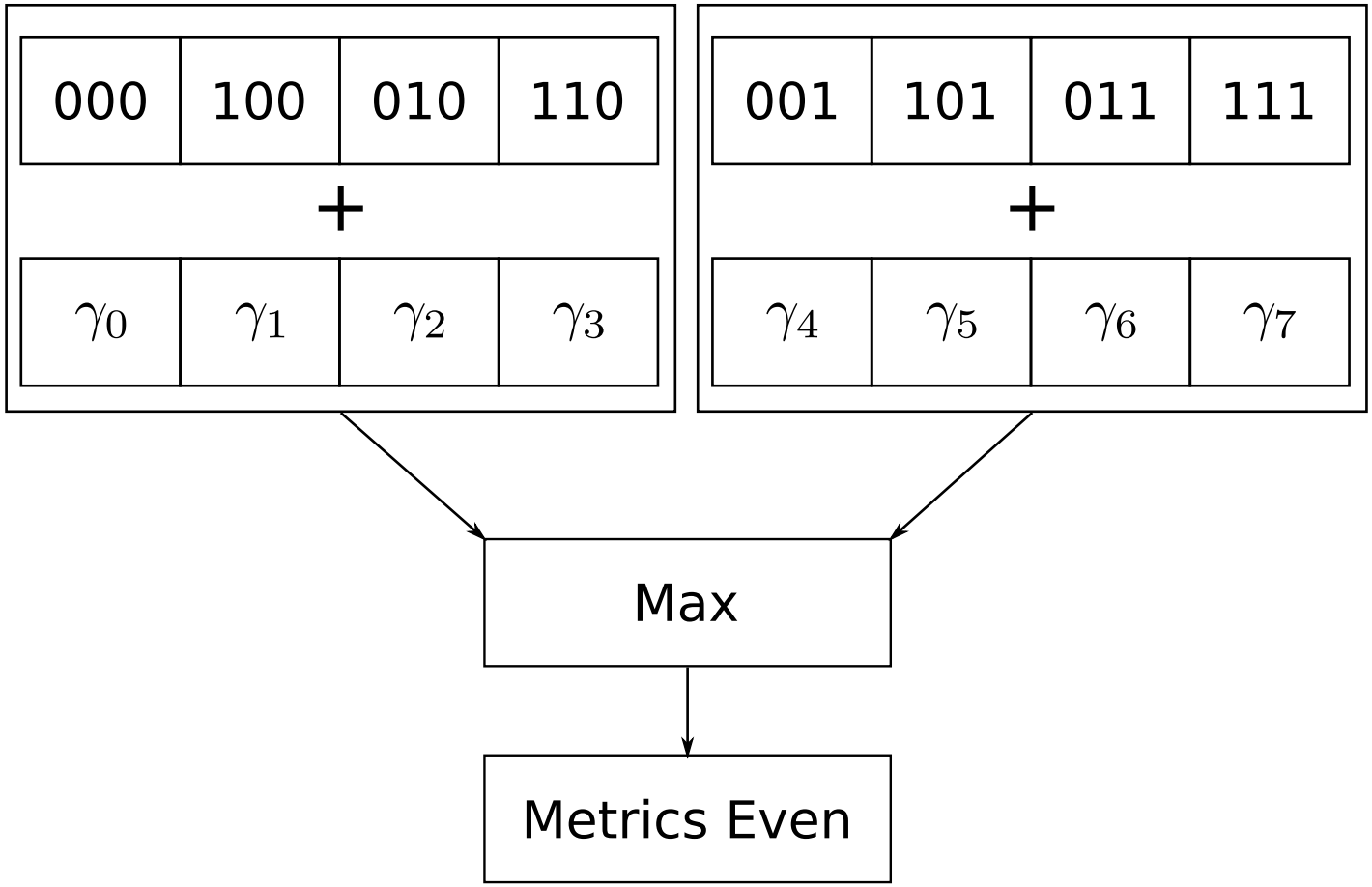
The Add-Compare-Select Butterfly



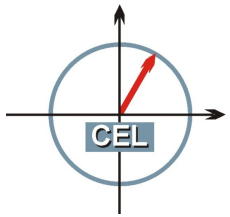


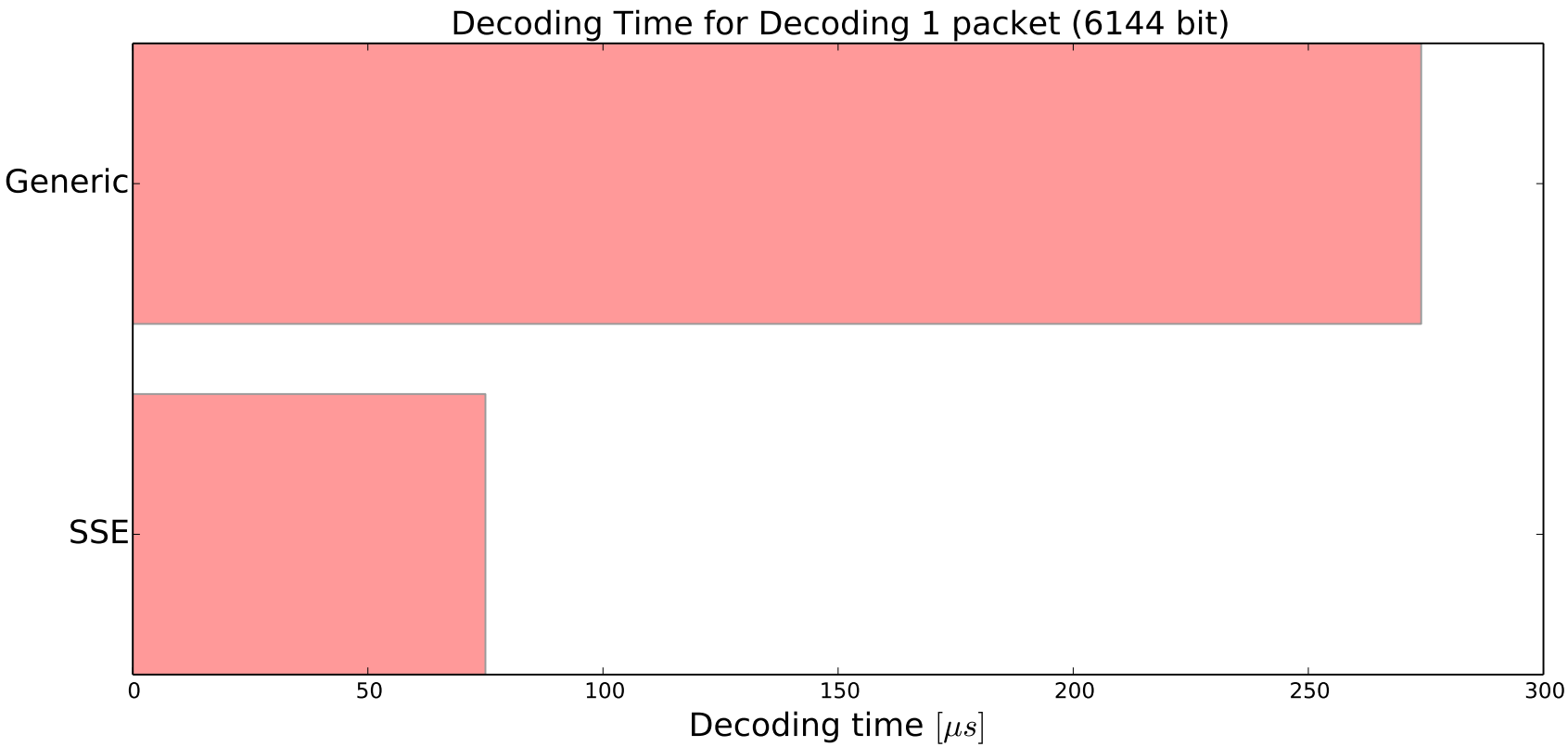
The Add-Compare-Select Operation



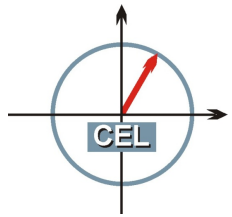


Now for some real numbers



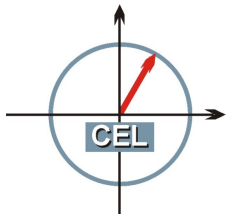


Conclusion



- SSE (float) is approx. 3.5x faster than standard C
- Memory constraints lead to reduced genericity
- Optimized Decoders are possible while a fair amount of flexibility is retained

That's all folks



- **Blog**:

  spectrejan.blogspot.com
- **GITHUB**:

  www.github.com/spectrejan
- **Email**:

  kraemer.jn@gmail.com