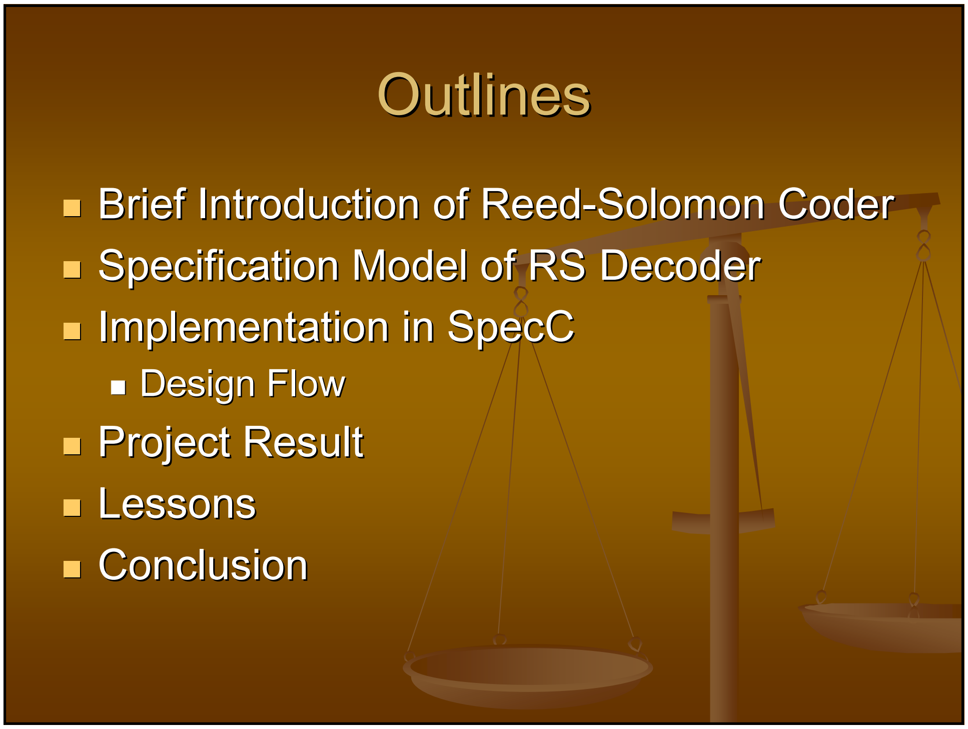
SpecC Implementation of Reed-Solomon Decoder

Jun Ho Bahn, *jbahn@uci.edu*
EECS



Outlines

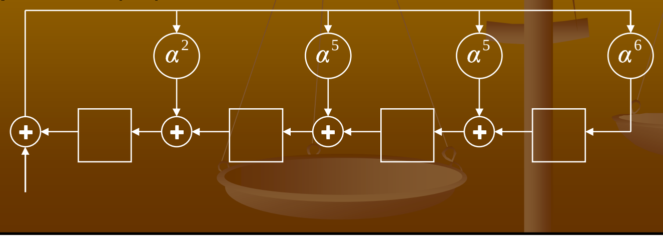
- Brief Introduction of Reed-Solomon Coder
- Specification Model of RS Decoder
- Implementation in SpecC
  - Design Flow
- Project Result
- Lessons
- Conclusion

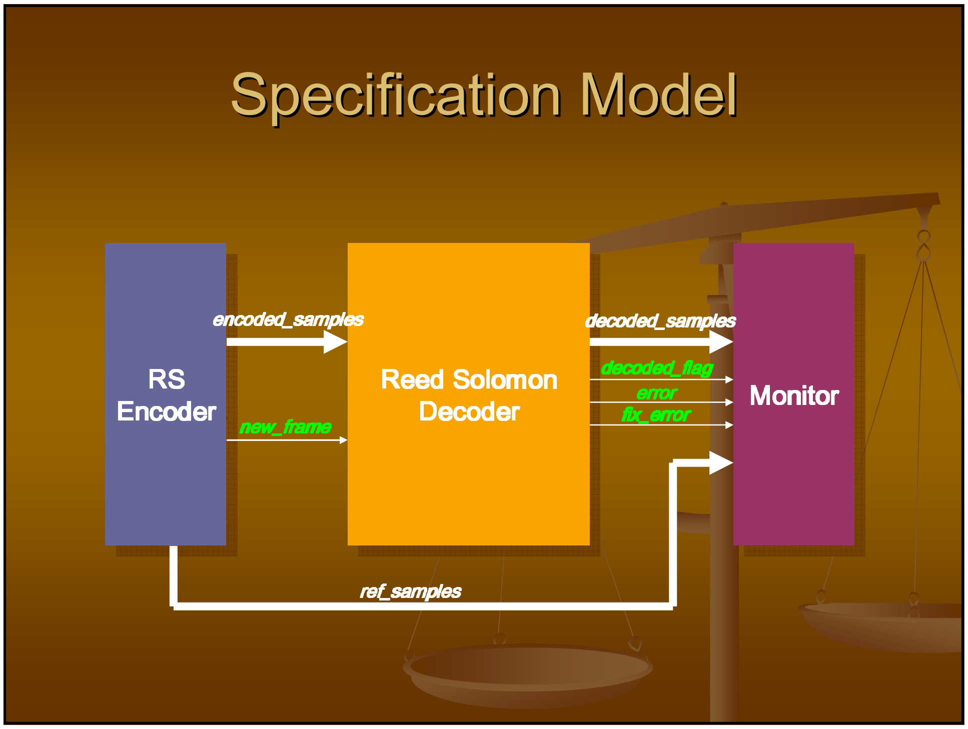
Brief Introduction of Reed-Solomon Coder

- Error Detection & Correction Channel Coder
  - FEC(forward error correction) coder
  - Extention of BCH coder
  - Cyclic Redundancy Coder
- Broadly used for digital VCR, CD, DVD and DTV
- *RS(n, k)*
  - *n* : code word length
  - *k* : information code word length
  - *n – k* : error check word length
  - max *(n-k)* word error detection & *(n-k)/2* word error correction

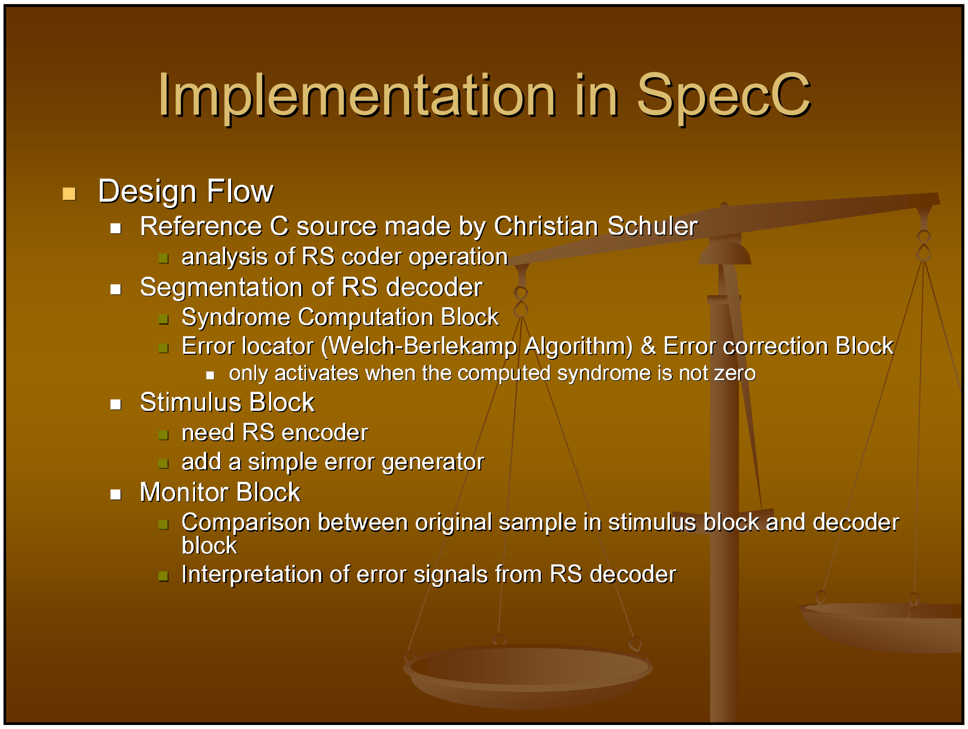
Brief Introduction (cont’d)

- Galois Field, GF(q)
  - Set of Finite number of Elements capable of 4 fundamental Arithmetics
  - Similar Characteristics with Modulo-q
- Example
  - RS(7, 3)
    - generator polynomial $g(X) = X^4 + \alpha^2 X^3 + \alpha^5 X^2 + \alpha^5 X + \alpha^6$





Specification Model

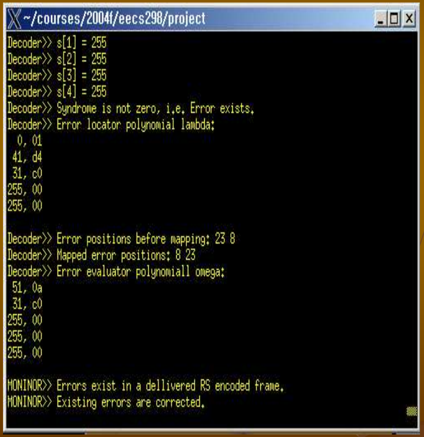


Implementation in SpecC

- Design Flow
  - Reference C source made by Christian Schuler
    - analysis of RS coder operation
  - Segmentation of RS decoder
    - Syndrome Computation Block
    - Error locator (Welch-Berlekamp Algorithm) & Error correction Block
      - only activates when the computed syndrome is not zero
  - Stimulus Block
    - need RS encoder
    - add a simple error generator
  - Monitor Block
    - Comparison between original sample in stimulus block and decoder block
    - Interpretation of error signals from RS decoder

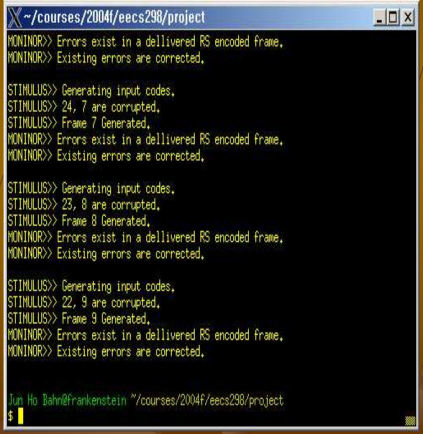
Project Results

- with Debugging Information



```
Decoder>> s[1] = 255
Decoder>> s[2] = 255
Decoder>> s[3] = 255
Decoder>> s[4] = 255
Decoder>> Syndrome is not zero, i.e. Error exists.
Decoder>> Error locator polynomial lambda:
  0, 01
 41, d4
 31, c0
255, 00
255, 00

Decoder>> Error positions before mapping: 23 8
Decoder>> Mapped error positions: 8 23
Decoder>> Error evaluator polynomiall omega:
 51, 0a
 31, c0
255, 00
255, 00
255, 00

MONINOR>> Errors exist in a dellivered RS encoded frame.
MONINOR>> Existing errors are corrected.
```

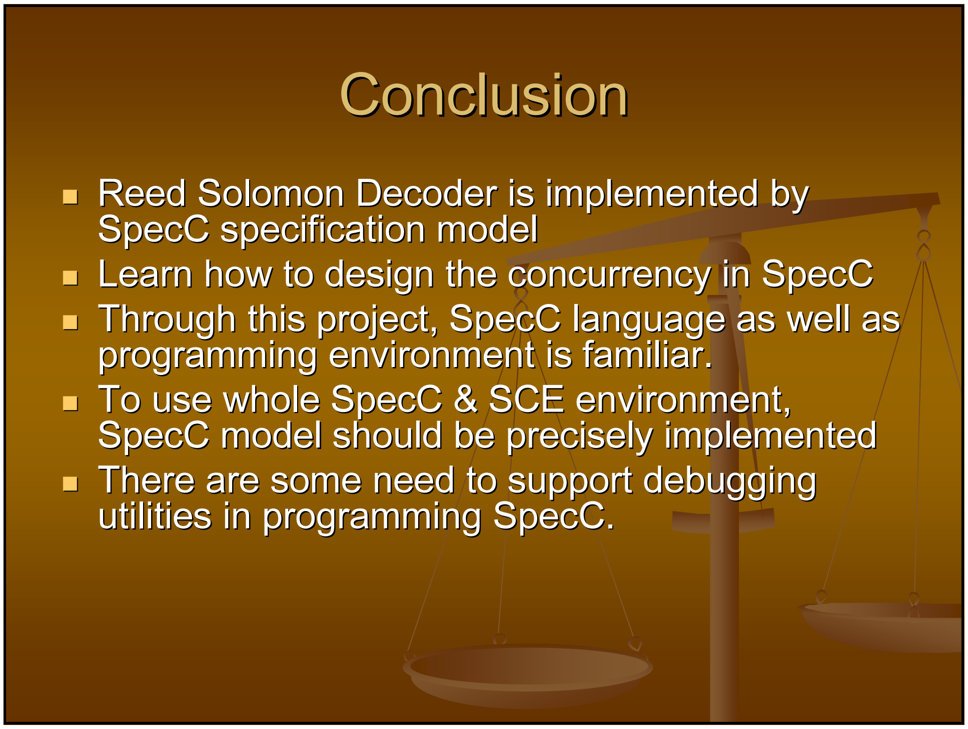
- w/o Debugging Information



```
MONINOR>> Errors exist in a dellivered RS encoded frame.
MONINOR>> Existing errors are corrected.

STIMULUS>> Generating input codes.
STIMULUS>> 24, 7 are corrupted.
STIMULUS>> Frame 7 Generated.
MONINOR>> Errors exist in a dellivered RS encoded frame.
MONINOR>> Existing errors are corrected.

STIMULUS>> Generating input codes.
STIMULUS>> 23, 8 are corrupted.
STIMULUS>> Frame 8 Generated.
MONINOR>> Errors exist in a dellivered RS encoded frame.
MONINOR>> Existing errors are corrected.

STIMULUS>> Generating input codes.
STIMULUS>> 22, 9 are corrupted.
STIMULUS>> Frame 9 Generated.
MONINOR>> Errors exist in a dellivered RS encoded frame.
MONINOR>> Existing errors are corrected.


Jun Ho Bahn@frankenstein ~/courses/2004f/eecs298/project
$
```

Lessons

- Concurrency is very attractive point in SpecC
- Similar effort should be needed to implement low-level of SpecC model in comparison with other RTL description such as Verilog or VHDL
- No debugger in SpecC



Conclusion

- Reed Solomon Decoder is implemented by SpecC specification model
- Learn how to design the concurrency in SpecC
- Through this project, SpecC language as well as programming environment is familiar.
- To use whole SpecC & SCE environment, SpecC model should be precisely implemented
- There are some need to support debugging utilities in programming SpecC.