Exam Conditions:

This is a take home exam. Your answers must be your own work.
No collaboration with other students in this class or any other person is allowed.
Any violation (receiving any outside help from anyone) may result in an X grade for this class.

You must submit a written document describing your work and the solutions. The written copy of the document must include a printout of the MATLAB code used. The MATLAB code software used to generate the solutions must also be e-mailed to Dr. Bazuin and be fully executable. Provide comments in your MATLAB code to describe what you are doing in the code and why the sections of code exist. Provide all supporting MATLAB functions required to make your main routines execute.

As a reminder:

WMU Codes, Policies, Processes and Procedures:

WMU STUDENT ACADEMIC CONDUCT POLICY

Students are responsible for making themselves aware of and understanding the University policies and procedures that pertain to Academic Honesty. These policies include cheating, fabrication, falsification and forgery, multiple submission, plagiarism, complicity and computer misuse.

The academic policies addressing Student Rights and Responsibilities can be found in the Graduate Catalog at http://catalog.wmich.edu/content.php?catoid=25&navoid=1030.

If there is reason to believe you have been involved in academic dishonesty, you will be referred to the Office of Student Conduct. You will be given the opportunity to review the charge(s) and if you believe you are not responsible, you will have the opportunity for a hearing. You should consult with your instructor if you are uncertain about an issue of academic honesty prior to the submission of an assignment or test.

Exam Submission:

You may submit your exam to me prior to 2:30 pm on 12 December.

The written exam document must be delivered to me prior to 7:00 pm on 27 April. If you are submitting by e-mail; the written document must be in either MS Word or Adobe Acrobat compatible *.pdf.

The MATLAB code must be submitted by e-mail in separate *.m files and be executable. All e-mail exam elements must be received in the WMU computer network (e-mail message time and date marking) by to 2:30 pm on 12 December.

It is the student’s responsibility to ensure on timely delivery of all materials.
Design and Comparative Performance Analysis of Three Types of Filter-Decimation and Interpolation-Filter Architectures: (1) Cascaded Half-Band Filters, (2) CIC Filters, and (3) a Generalized Polyphase Filters.

You are expected to submit both a written document and all MATLAB code. The written document should contain all required design information, designs and listings required (part ‘a’ for each question). It should also provide a summary (with plots) of the performance results achieved for each question (parts b and d).

(1) [25 pts] Cascaded Half-Band Filters

Similar to Exam #2 using direct and polyphase half band filters.

(1.1) Interpolation

Modified and/or use previously provided code to (1) generate ASK symbol values, (2) define a square root Nyquist filter for sample to symbol interpolation, and (3) use an interpolate by 5 with the Nyquist filter to generate 5 samples per symbol ASK time samples.

Test Signal Specifications

<table>
<thead>
<tr>
<th>ASK Modulated Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample Rate</td>
</tr>
<tr>
<td>Symbol Rate</td>
</tr>
<tr>
<td>Symbol Bandwidth</td>
</tr>
<tr>
<td>Roll-off Bandwidth</td>
</tr>
<tr>
<td>Interpolation filter</td>
</tr>
</tbody>
</table>

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample Rate Output</td>
</tr>
<tr>
<td>Symbol Bandwidth</td>
</tr>
<tr>
<td>Pass-band Ripple</td>
</tr>
<tr>
<td>Stop-band Attenuation</td>
</tr>
</tbody>
</table>

Half-Band Interpolator Specifications

Interpolate the test signal by a factor of 32 using a cascade of “interpolate by 2 and minimum coefficient half-band filters” stages as describe in Section 8.6 of the text (p. 214) and shown in Fig. 8.21 (p. 218). Your design should meet the following filter specification for each individual half-band filter used. Verify the simulation performance using (a) the test signal and (b) an impulse response. Show the frequency responses for each successive stage of interpolate-half band filters. For (b), the impulse response, describe why the shape of the impulse response interpolated spectrum would be expected based on the signal processing performed.

HB Filter Specifications

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample Rate Output</td>
</tr>
<tr>
<td>Symbol Bandwidth</td>
</tr>
<tr>
<td>Pass-band Ripple</td>
</tr>
<tr>
<td>Stop-band Attenuation</td>
</tr>
</tbody>
</table>
(1.2) Decimation
Create the mirror image of the interpolator to perform filter decimation by a factor of 32 using a cascade of “half-band filter decimate by 2” stages. The output filter specifications are “the inverse” of those used for interpolation.

(1.3) Apply a square root Nyquist filter to generate the test signal input and then apply another square root Nyquist filter to the part 1.2 output. Make an “eye” diagram. You should be able to observe the original ASK symbol levels.

Verify your simulation using (a) an impulse response and (b) the output of the simulation from part 1.1 above. Show the frequency responses for each successive stage of half band filter-decimator. Compare the ASK test signal time sample input to the decimated output.

For problems 1.1 and 1.2 and 1.3:

a) Provide a block diagram of the design and a listing of the critical design elements and relevant equations (with important design parameters and/or filter specifications).

b) Generate the MATLAB code and provide spectral plots for all filters required.

c) Generate the MATLAB code to perform the signal processing blocks and operations defined by the specific option.

d) Use provided test signals and generate appropriate test signals to validate MATLAB simulations and demonstrate correct behavior.

e) Provide any specific answers or tables that are useful or even necessary to describe the performance results.
(2) [25 pts] Cascaded Integrator Comb Filters – Hogenauer Filters.

(2.1) Interpolation

Modified and/or use previously provided code to (1) generate ASK symbol values, (2) define a square root Nyquist filter for sample to symbol interpolation, and (3) use an interpolate by 4 with the Nyquist filter to generate 4 samples per symbol ASK time samples.

**Test Signal Specifications**

<table>
<thead>
<tr>
<th>ASK Modulated Signal</th>
</tr>
</thead>
</table>
| Sample Rate          | 4  
| Symbol Rate          | 1  
| Symbol Bandwidth     | 1.4 (-0.7 to 0.7)  
| Roll-off Bandwidth   | 0.4  
| Interpolation filter | Nyquist  

**Cascaded Integrator Comb Interpolation**

Interpolate the test signal by a factor of 100 using a five-stage Hogenauer interpolator as describe in Section 11.4 of the text (p. 341) and with a similar (expanded) design to Fig. 11.17 (p. 345). Use appropriate “integer processing” for the CIC stages (a floating point implementation may overflow the mantissa and not work correctly). Assume that the input test signal is a 16-bit two’s complement number with appropriate scaling to use the full numerical range. The output of your design should meet the following specification. Verify the simulation performance using (a) the test signal and (b) an impulse response. Provide gain normalization for the output signals and define the number of bits required if all stages used the same bit precision.

**Interpolator Filter Specifications**

<table>
<thead>
<tr>
<th>Hogenauer Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Sample Rate</td>
</tr>
<tr>
<td>Output Sample Rate</td>
</tr>
<tr>
<td>Symbol Bandwidth</td>
</tr>
</tbody>
</table>

Option/extra credit: add a “clean-up” filter, inverse sinc function filter, between the test signal generator and the CIC interpolator (pre-emphasis at a lower sample rate).

(2.2) Decimation

**Cascaded Integrator Comb Decimation**

Create the mirror image of the interpolator to perform filter decimation by a factor of 100 using a five-stage Hogenauer. Use appropriate “integer processing” for the CIC stages (a floating point implementation may overflow the mantissa and not work correctly). Assume that the input test
signal is a 16-bit two’s complement number with appropriate scaling to use the full numerical range. The output filter specifications are:

**Prototype Filter Specifications**

<table>
<thead>
<tr>
<th>Filter</th>
<th>Input Sample Rate</th>
<th>Output Sample Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hogenauer Filter</td>
<td>400</td>
<td>4</td>
</tr>
</tbody>
</table>

Verify your simulation using (a) an impulse response and (b) the output of the simulation from part 2.1 above. Provide gain normalization for the output signals.

Option/extra credit: add a “clean-up” filter, inverse sinc function filter, after the CIC decimator (post-filter at a lower sample rate).

(2.3) Apply a square root Nyquist filter to the part 2.2 output and make an “eye” diagram, you should be able to observe the original ASK modulation and symbol values.

For problems 2.1 and 2.2 and 2.3:

a) Provide a block diagram of the design and a listing of the critical design elements and relevant equations (with important design parameters and/or filter specifications).

b) Generate the MATLAB code and provide spectral plots for all filters required.

c) Generate the MATLAB code to perform the signal processing blocks and operations defined by the specific option.

d) Generate appropriate test signals to validate MATLAB simulations and demonstrate correct behavior.

e) Provide any specific answers or tables that are useful or even necessary to describe the performance results.
(3) [20 pts] Generalized Polyphase Synthesizer

For this exam, you will develop the design, MATLAB simulations, and provide the performance results of two design approaches of the polyphase synthesizer requirements defined in the paper:


(3.1) Direct Upconversion (interpolate by 48, Nyquist filter and mix) as shown here
(3.2) Generalized polyphase synthesis design as described in the paper and on-line

The implementation is mathematically defined in the Generalized Synthesis paper provide in the class notes on-line and a MATLAB example using different rates, filter, etc. has been provided.
For problems 3.1 and 3.2:

a) Provide a block diagram of the design (as shown above?) and a listing of the critical design elements and relevant equations (with important design parameters and/or filter specifications).

b) Generate the MATLAB code and provide spectral plots for all filters required.

c) Generate the MATLAB code to perform the signal processing blocks and operations defined by the specific option.

d) Generate appropriate test signals to validate MATLAB simulations and demonstrate correct behavior.