CDC6600/CDC7600 Simulator

Group Leader: Aaron Easter

Group Members:

Westin Sykes

Aaron Easter

By: Westin Sykes

**Problem Description**

This project included the simulation of the CDC6600 and CDC7600 in either C or C++. This required the design and implementation of the instruction pipeline, scoreboard, functional units, and a logging system which was capable of writing a timing diagram of the code the simulator executes.

The first step in implementing the simulator was to design code that could be abstracted to both the CDC6600 and CDC7600. We decided a single class for the functional units could be used. A pipeline was built in to this functional unit class, with the ability to make the segment time equal to the execution time effectively nullifying the pipeline for the CDC6600. This allowed for the functional unit code to be easily abstracted to each system. These functional units had a “clockTick” function which would push instructions down each one’s own pipeline as long as there was no conflict requiring the instruction to not be executed after being issued.

The scoreboard object originally was written with abstraction in mind, but it eventually was clear that this was not the best plan, and making two different scoreboards was necessary. The scoreboard for the 6600 and 7600 were very similar with only a few major differences in which functional units it utilized and how it chose which one to use for a given instruction.

The instruction pipeline was identical for both the 6600 and 7600. It has the functionality of converting invalid instructions after a long instruction to NO-OPs so the code for the functional units would be slightly simpler. It uses a similar “clockTick” function as the functional units which pushes instructions down the pipe assuming the pipeline had not been stopped due to a conflict.

The logging system used to create the timing diagrams, known as “TimingDiagram” in the code, is a public object associated with the simulation. This object stores each issued instruction and the events along its lifetime. These events include issue, start, result, unit ready, fetch and store. It accessed a member variable called “clockTickCount” which stored the number of times the “clockTick” function had been called since the beginning of the simulation to log the time of each event as they occur. Using these main parts along with some other code to pull it all together, these pieces interacted nicely together.

**Results**

**Y = AX2 + B**

-CDC6600

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Word # | Instruction | Semantics |  | Issue | Start | Result | Unit | Fetch | Store |
|  |  |  |  |  |  |  |  |  |  |
| N1 | A1 = A1 + K1 | Fetch X |  | 1 | 1 | 4 | 4 | 9 |  |
|  | A2 = A2 + K2 | Fetch A |  | 3 | 3 | 6 | 6 | 11 |  |
| N2 | X0 = X1 \* X1 | Form X^2 |  | 9 | 9 | 19 | 19 |  |  |
|  | X6 = X0 \* X2 | Form AX^2 |  | 10 | 19 | 29 | 29 |  |  |
|  | A3 = A3 + K3 | Fetch B |  | 11 | 11 | 14 | 14 | 19 |  |
| N3 | X7 = X6 + X3 | Form Y |  | 17 | 29 | 33 | 33 |  |  |
|  | A7 = A7 + K7 | Store Y |  | 18 | 33 | 36 | 36 |  | 41 |

-CDC7600

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Word # | Instruction | Semantics |  | Issue | Start | Result | Unit | Fetch | Store |
|  |  |  |  |  |  |  |  |  |  |
| N1 | A1 = A1 + K1 | Fetch X |  | 1 | 1 | 3 | 2 | 7 |  |
|  | A2 = A2 + K2 | Fetch A |  | 3 | 3 | 5 | 4 | 9 |  |
| N2 | X0 = X1 \* X1 | Form X^2 |  | 7 | 7 | 12 | 9 |  |  |
|  | X6 = X0 \* X2 | Form AX^2 |  | 9 | 12 | 17 | 14 |  |  |
|  | A3 = A3 + K3 | Fetch B |  | 10 | 10 | 12 | 11 | 16 |  |
| N3 | X7 = X6 + X3 | Form Y |  | 13 | 17 | 21 | 18 |  |  |
|  | A7 = A7 + K7 | Store Y |  | 14 | 21 | 23 | 22 |  | 27 |

**Y = AX2 + BX + C**

-CDC6600

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Word # | Instruction | Semantics |  | Issue | Start | Result | Unit | Fetch | Store |
|  |  |  |  |  |  |  |  |  |  |
| N1 | A1 = A1 + K1 | Fetch X |  | 1 | 1 | 4 | 4 | 9 |  |
|  | A2 = A2 + K2 | Fetch A |  | 3 | 3 | 6 | 6 | 11 |  |
| N2 | X0 = X1 \* X1 | Form X^2 |  | 9 | 9 | 19 | 19 |  |  |
|  | A3 = A3 + K3 | Fetch B |  | 10 | 10 | 13 | 13 | 18 |  |
|  | X6 = X0 \* X2 | Form AX^2 |  | 12 | 19 | 29 | 29 |  |  |
| N3 | A4 = A4 + K4 | Fetch C |  | 17 | 17 | 20 | 20 | 25 |  |
|  | X3 = X3 \* X1 | Form BX |  | 20 | 20 | 30 | 30 |  |  |
|  | X5 = X6 + X3 | Form AX^2 + BX |  | 21 | 30 | 34 | 34 |  |  |
| N4 | X7 = X5 + X4 | Form Y |  | 34 | 34 | 38 | 38 |  |  |
|  | A7 = A7 + K5 | Store Y |  | 35 | 38 | 41 | 41 |  | 46 |

-CDC7600

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Word # | Instruction | Semantics |  | Issue | Start | Result | Unit | Fetch | Store |
|  |  |  |  |  |  |  |  |  |  |
| N1 | A1 = A1 + K1 | Fetch X |  | 1 | 1 | 3 | 2 | 7 |  |
|  | A2 = A2 + K2 | Fetch A |  | 3 | 3 | 5 | 4 | 9 |  |
| N2 | X0 = X1 \* X1 | Form X^2 |  | 7 | 7 | 12 | 9 |  |  |
|  | A3 = A3 + K3 | Fetch B |  | 8 | 8 | 10 | 9 | 14 |  |
|  | X6 = X0 \* X2 | Form AX^2 |  | 10 | 12 | 17 | 14 |  |  |
| N3 | A4 = A4 + K4 | Fetch C |  | 13 | 13 | 15 | 14 | 19 |  |
|  | X3 = X3 \* X1 | Form BX |  | 15 | 15 | 20 | 17 |  |  |
|  | X5 = X6 + X3 | Form AX^2 + BX |  | 16 | 20 | 24 | 21 |  |  |
| N4 | X7 = X5 + X4 | Form Y |  | 21 | 24 | 28 | 25 |  |  |
|  | A7 = A7 + K5 | Store Y |  | 22 | 28 | 30 | 29 |  | 34 |

**Y = AX2 + BX + C <X and Y are vectors, n=5>**

-CDC6600

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Word # | Instruction | Semantics |  | Issue | Start | Result | Unit | Fetch | Store |
|  |  |  |  |  |  |  |  |  |  |
| N1 | A5 = A5 + K5 | Fetch n |  | 1 | 1 | 4 | 4 | 9 |  |
|  | A2 = A2 + K2 | Fetch A |  | 3 | 3 | 6 | 6 | 11 |  |
| N2 | B2 = X5 + B0 | Move n to B2 |  | 9 | 9 | 12 | 12 |  |  |
|  | A3 = A3 + K3 | Fetch B |  | 10 | 10 | 13 | 13 | 18 |  |
| N3 | A4 = A4 + K4 | Fetch C |  | 17 | 17 | 20 | 20 | 25 |  |
|  | B1 = B0 + K0 | Set B1 = 1 |  | 19 | 19 | 22 | 22 |  |  |
| N4 | A1 = B2 + K1 | Fetch X[n] |  | 25 | 25 | 28 | 28 | 33 |  |
|  | X0 = X1 \* X1 | Form X^2 |  | 27 | 27 | 37 | 37 |  |  |
|  | X5 = X3 \* X1 | Form BX |  | 28 | 28 | 38 | 38 |  |  |
| N5 | X6 = X0 \* X2 | Form AX^2 |  | 38 | 38 | 48 | 48 |  |  |
|  | X5 = X5 + X4 | Form BX + C |  | 39 | 39 | 43 | 43 |  |  |
|  | X7 = X6 + X5 | Form Y[n] |  | 43 | 48 | 52 | 52 |  |  |
| N6 | A7 = B2 + K7 | Store Y[n] |  | 45 | 52 | 55 | 55 |  | 60 |
|  | B2 = B2 - B1 | Decrement n |  | 46 | 46 | 49 | 49 |  |  |
| N7 | Branch |  |  | 50 | 50 | 58 | 58 |  |  |
| N4 | A1 = B2 + K1 | Fetch X[n] |  | 59 | 59 | 62 | 62 | 67 |  |
|  | X0 = X1 \* X1 | Form X^2 |  | 61 | 61 | 71 | 71 |  |  |
|  | X5 = X3 \* X1 | Form BX |  | 62 | 62 | 72 | 72 |  |  |
| N5 | X6 = X0 \* X2 | Form AX^2 |  | 72 | 72 | 82 | 82 |  |  |
|  | X5 = X5 + X4 | Form BX + C |  | 73 | 73 | 77 | 77 |  |  |
|  | X7 = X6 + X5 | Form Y[n] |  | 77 | 82 | 86 | 86 |  |  |
| N6 | A7 = B2 + K7 | Store Y[n] |  | 79 | 86 | 89 | 89 |  | 94 |
|  | B2 = B2 - B1 | Decrement n |  | 80 | 80 | 83 | 83 |  |  |
| N7 | Branch |  |  | 84 | 84 | 92 | 92 |  |  |
| N4 | A1 = B2 + K1 | Fetch X[n] |  | 93 | 93 | 96 | 96 | 101 |  |
|  | X0 = X1 \* X1 | Form X^2 |  | 95 | 95 | 105 | 105 |  |  |
|  | X5 = X3 \* X1 | Form BX |  | 96 | 96 | 106 | 106 |  |  |
| N5 | X6 = X0 \* X2 | Form AX^2 |  | 106 | 106 | 116 | 116 |  |  |
|  | X5 = X5 + X4 | Form BX + C |  | 107 | 107 | 111 | 111 |  |  |
|  | X7 = X6 + X5 | Form Y[n] |  | 111 | 116 | 120 | 120 |  |  |
| N6 | A7 = B2 + K7 | Store Y[n] |  | 113 | 120 | 123 | 123 |  | 128 |
|  | B2 = B2 - B1 | Decrement n |  | 114 | 114 | 117 | 117 |  |  |
| N7 | Branch |  |  | 118 | 118 | 126 | 126 |  |  |
| N4 | A1 = B2 + K1 | Fetch X[n] |  | 127 | 127 | 130 | 130 | 135 |  |
|  | X0 = X1 \* X1 | Form X^2 |  | 129 | 129 | 139 | 139 |  |  |
|  | X5 = X3 \* X1 | Form BX |  | 130 | 130 | 140 | 140 |  |  |
| N5 | X6 = X0 \* X2 | Form AX^2 |  | 140 | 140 | 150 | 150 |  |  |
|  | X5 = X5 + X4 | Form BX + C |  | 141 | 141 | 145 | 145 |  |  |
|  | X7 = X6 + X5 | Form Y[n] |  | 145 | 150 | 154 | 154 |  |  |
| N6 | A7 = B2 + K7 | Store Y[n] |  | 147 | 154 | 157 | 157 |  | 162 |
|  | B2 = B2 - B1 | Decrement n |  | 148 | 148 | 151 | 151 |  |  |
| N7 | Branch |  |  | 152 | 152 | 160 | 160 |  |  |
| N4 | A1 = B2 + K1 | Fetch X[n] |  | 161 | 161 | 164 | 164 | 169 |  |
|  | X0 = X1 \* X1 | Form X^2 |  | 163 | 163 | 173 | 173 |  |  |
|  | X5 = X3 \* X1 | Form BX |  | 164 | 164 | 174 | 174 |  |  |
| N5 | X6 = X0 \* X2 | Form AX^2 |  | 174 | 174 | 184 | 184 |  |  |
|  | X5 = X5 + X4 | Form BX + C |  | 175 | 175 | 179 | 179 |  |  |
|  | X7 = X6 + X5 | Form Y[n] |  | 179 | 184 | 188 | 188 |  |  |
| N6 | A7 = B2 + K7 | Store Y[n] |  | 181 | 188 | 191 | 191 |  | 196 |
|  | B2 = B2 - B1 | Decrement n |  | 182 | 182 | 185 | 185 |  |  |
| N7 | Branch |  |  | 186 | 186 | 194 | 194 |  |  |
| N4 | A1 = B2 + K1 | Fetch X[n] |  | 195 | 195 | 198 | 198 | 203 |  |
|  | X0 = X1 \* X1 | Form X^2 |  | 197 | 197 | 207 | 207 |  |  |
|  | X5 = X3 \* X1 | Form BX |  | 198 | 198 | 208 | 208 |  |  |
| N5 | X6 = X0 \* X2 | Form AX^2 |  | 208 | 208 | 218 | 218 |  |  |
|  | X5 = X5 + X4 | Form BX + C |  | 209 | 209 | 213 | 213 |  |  |
|  | X7 = X6 + X5 | Form Y[n] |  | 213 | 218 | 222 | 222 |  |  |
| N6 | A7 = B2 + K7 | Store Y[n] |  | 215 | 222 | 225 | 225 |  | 230 |
|  | B2 = B2 - B1 | Decrement n |  | 216 | 216 | 219 | 219 |  |  |

-CDC7600

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Word # | Instruction | Semantics |  | Issue | Start | Result | Unit | Fetch | Store |
|  |  |  |  |  |  |  |  |  |  |
| N1 | A5 = A5 + K5 | Fetch n |  | 1 | 1 | 3 | 2 | 7 |  |
|  | A2 = A2 + K2 | Fetch A |  | 3 | 3 | 5 | 4 | 9 |  |
| N2 | B2 = X5 + B0 | Move n to B2 |  | 7 | 7 | 9 | 8 |  |  |
|  | A3 = A3 + K3 | Fetch B |  | 8 | 8 | 10 | 9 | 14 |  |
| N3 | A4 = A4 + K4 | Fetch C |  | 13 | 13 | 15 | 14 | 19 |  |
|  | B1 = B0 + K0 | Set B1 = 1 |  | 15 | 15 | 17 | 16 |  |  |
| N4 | A1 = B2 + K1 | Fetch X[n] |  | 19 | 19 | 21 | 20 | 25 |  |
|  | X0 = X1 \* X1 | Form X^2 |  | 21 | 21 | 26 | 23 |  |  |
|  | X5 = X3 \* X1 | Form BX |  | 23 | 23 | 28 | 25 |  |  |
| N5 | X6 = X0 \* X2 | Form AX^2 |  | 26 | 26 | 31 | 28 |  |  |
|  | X5 = X5 + X4 | Form BX + C |  | 27 | 28 | 32 | 29 |  |  |
|  | X7 = X6 + X5 | Form Y[n] |  | 29 | 32 | 36 | 33 |  |  |
| N6 | A7 = B2 + K7 | Store Y[n] |  | 32 | 36 | 38 | 37 |  | 42 |
|  | B2 = B2 - B1 | Decrement n |  | 37 | 37 | 39 | 38 |  |  |
| N7 | Branch |  |  | 40 | 40 | 43 | 41 |  |  |
| N4 | A1 = B2 + K1 | Fetch X[n] |  | 49 | 49 | 51 | 50 | 55 |  |
|  | X0 = X1 \* X1 | Form X^2 |  | 51 | 51 | 56 | 53 |  |  |
|  | X5 = X3 \* X1 | Form BX |  | 53 | 53 | 58 | 55 |  |  |
| N5 | X6 = X0 \* X2 | Form AX^2 |  | 56 | 56 | 61 | 58 |  |  |
|  | X5 = X5 + X4 | Form BX + C |  | 57 | 58 | 62 | 59 |  |  |
|  | X7 = X6 + X5 | Form Y[n] |  | 59 | 62 | 66 | 63 |  |  |
| N6 | A7 = B2 + K7 | Store Y[n] |  | 62 | 66 | 68 | 67 |  | 72 |
|  | B2 = B2 - B1 | Decrement n |  | 67 | 67 | 69 | 68 |  |  |
| N7 | Branch |  |  | 70 | 70 | 73 | 71 |  |  |
| N4 | A1 = B2 + K1 | Fetch X[n] |  | 79 | 79 | 81 | 80 | 85 |  |
|  | X0 = X1 \* X1 | Form X^2 |  | 81 | 81 | 86 | 83 |  |  |
|  | X5 = X3 \* X1 | Form BX |  | 83 | 83 | 88 | 85 |  |  |
| N5 | X6 = X0 \* X2 | Form AX^2 |  | 86 | 86 | 91 | 88 |  |  |
|  | X5 = X5 + X4 | Form BX + C |  | 87 | 88 | 92 | 89 |  |  |
|  | X7 = X6 + X5 | Form Y[n] |  | 89 | 92 | 96 | 93 |  |  |
| N6 | A7 = B2 + K7 | Store Y[n] |  | 92 | 96 | 98 | 97 |  | 102 |
|  | B2 = B2 - B1 | Decrement n |  | 97 | 97 | 99 | 98 |  |  |
| N7 | Branch |  |  | 100 | 100 | 103 | 101 |  |  |
| N4 | A1 = B2 + K1 | Fetch X[n] |  | 109 | 109 | 111 | 110 | 115 |  |
|  | X0 = X1 \* X1 | Form X^2 |  | 111 | 111 | 116 | 113 |  |  |
|  | X5 = X3 \* X1 | Form BX |  | 113 | 113 | 118 | 115 |  |  |
| N5 | X6 = X0 \* X2 | Form AX^2 |  | 116 | 116 | 121 | 118 |  |  |
|  | X5 = X5 + X4 | Form BX + C |  | 117 | 118 | 122 | 119 |  |  |
|  | X7 = X6 + X5 | Form Y[n] |  | 119 | 122 | 126 | 123 |  |  |
| N6 | A7 = B2 + K7 | Store Y[n] |  | 122 | 126 | 128 | 127 |  | 132 |
|  | B2 = B2 - B1 | Decrement n |  | 127 | 127 | 129 | 128 |  |  |
| N7 | Branch |  |  | 130 | 130 | 133 | 131 |  |  |
| N4 | A1 = B2 + K1 | Fetch X[n] |  | 139 | 139 | 141 | 140 | 145 |  |
|  | X0 = X1 \* X1 | Form X^2 |  | 141 | 141 | 146 | 143 |  |  |
|  | X5 = X3 \* X1 | Form BX |  | 143 | 143 | 148 | 145 |  |  |
| N5 | X6 = X0 \* X2 | Form AX^2 |  | 146 | 146 | 151 | 148 |  |  |
|  | X5 = X5 + X4 | Form BX + C |  | 147 | 148 | 152 | 149 |  |  |
|  | X7 = X6 + X5 | Form Y[n] |  | 149 | 152 | 156 | 153 |  |  |
| N6 | A7 = B2 + K7 | Store Y[n] |  | 152 | 156 | 158 | 157 |  | 162 |
|  | B2 = B2 - B1 | Decrement n |  | 157 | 157 | 159 | 158 |  |  |
| N7 | Branch |  |  | 160 | 160 | 163 | 161 |  |  |
| N4 | A1 = B2 + K1 | Fetch X[n] |  | 169 | 169 | 171 | 170 | 175 |  |
|  | X0 = X1 \* X1 | Form X^2 |  | 171 | 171 | 176 | 173 |  |  |
|  | X5 = X3 \* X1 | Form BX |  | 173 | 173 | 178 | 175 |  |  |
| N5 | X6 = X0 \* X2 | Form AX^2 |  | 176 | 176 | 181 | 178 |  |  |
|  | X5 = X5 + X4 | Form BX + C |  | 177 | 178 | 182 | 179 |  |  |
|  | X7 = X6 + X5 | Form Y[n] |  | 179 | 182 | 186 | 183 |  |  |
| N6 | A7 = B2 + K7 | Store Y[n] |  | 182 | 186 | 188 | 187 |  | 192 |
|  | B2 = B2 - B1 | Decrement n |  | 187 | 187 | 189 | 188 |  |  |

**Discussion**

As can be seen from the results above, the CDC7600 performs better than the CDC6600 in every case. For the first equation (Y = AX^2 + B), one of the biggest bottlenecks were true data dependencies from functional units being forced to wait for previous multiplication operations to write the result to the register so that it can be read by the next instruction. The CDC7600 was able to perform significantly better here because of a substantially sped-up multiplication operation, resulting in less waiting.

For the second equation (Y = AX^2 + Bx + C), the CSDC7600 was again able to substantially outperform the CDC6600 as a result of quicker operations, namely multiplication, and shorter memory access times. As a result, it didn't have to wait as long for the true data dependencies to be resolved.

The result of the third equation followed the same trend as the previous two with the CDC7600 noticeably outperforming the CDC6600; however, it was by a proportionally smaller amount than with the previous ones. The cause of this is the frequent branching that takes place while looping over the small number of instructions present, as well as the data dependencies in the instructions that are being iteratively executed.