The J Computer (Appendix)

Advay Mengle

6.111 Spring 2007 Final Project
T.A: Amir Hirsch
May 17, 2007

Table of Contents

A Simulation Waveforms

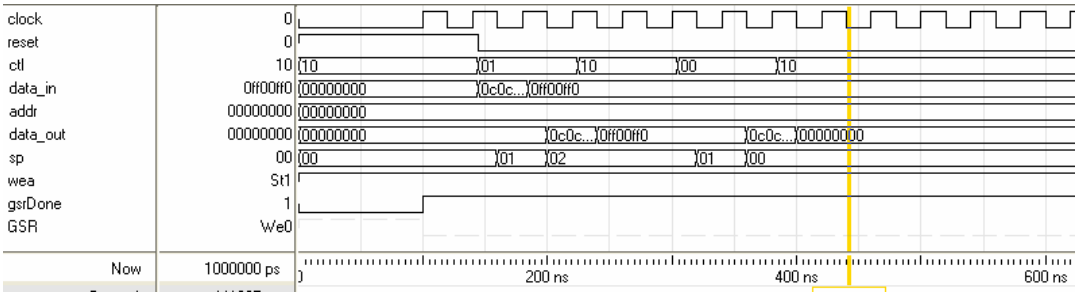

Figure 1 – Simple stack manipulation operations. The stack pointer (sp) moves through the address space of the stack manager's BRAM as 32-bit words are pushed and popped

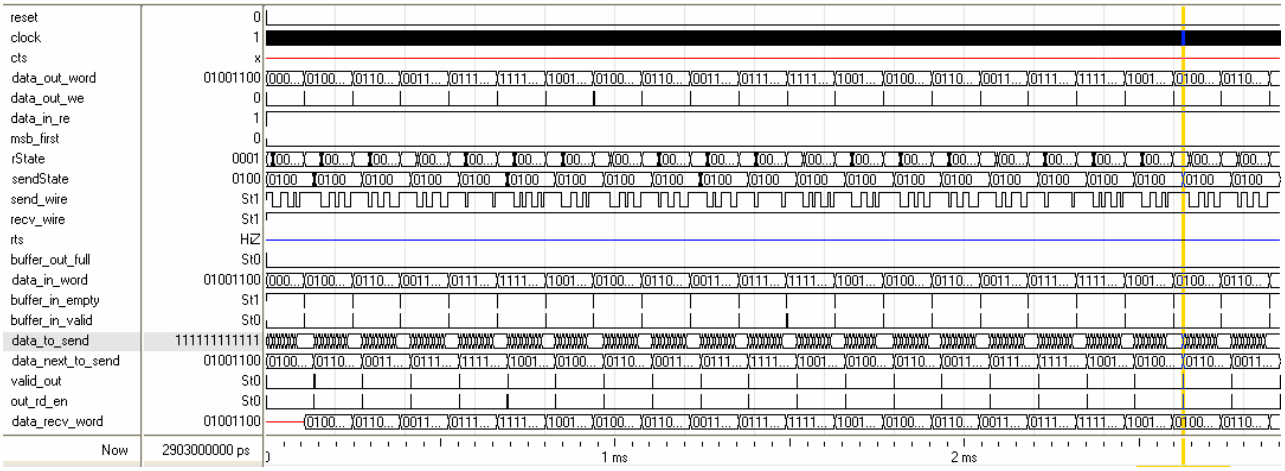

Figure 2 – Buffered RS-232 communication. Two RS232_UART modules were instantiated and connected to each other. A small block of data stored in one UART's send buffer is initially sent to the other, and then the data forever bounces between the two UARTs.

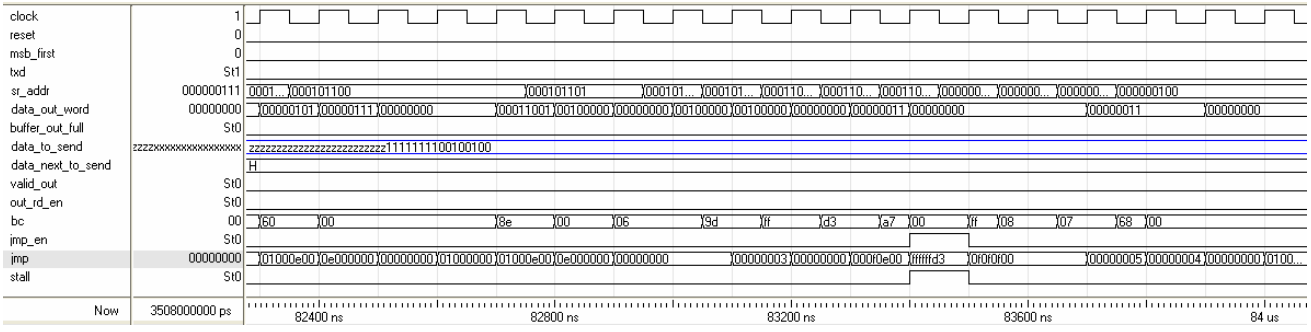

Figure 3 – Demonstration of a successful branch operation. At 83400 ns, the ifgt bytecode (opcode 0x9D, meaning "if greater than 0, then branch") causes the processor to stall and jump 45 bytes backwards (-45 è immediate literal 0xFFD3). Simulation of the entire loop can be seen in Figure 4.

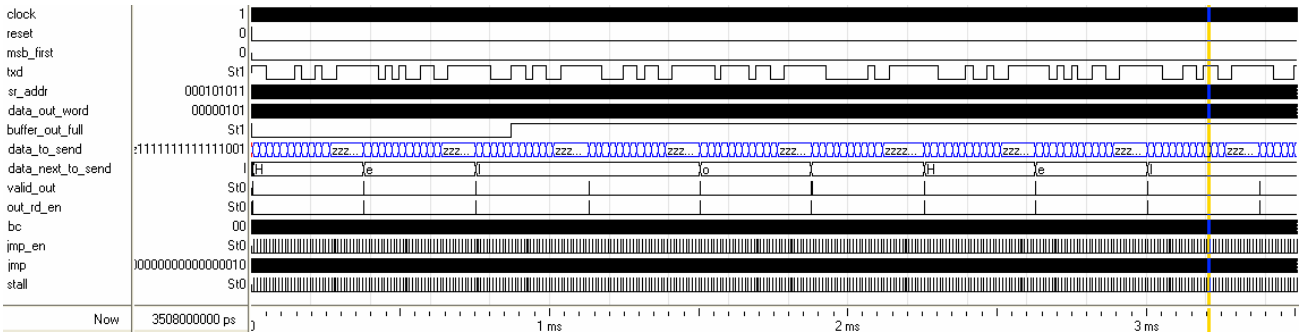

Figure 4 – Demonstration of a successful infinite loop. A single execution of the loop pushes the string "Hello" to the RS-232 send buffer. Because the transmission of characters over RS-232 occurs at a much slower rate than execution of bytecode, the procedure loops many times (once every time jmp_en is asserted) before even one character is sent and the RS-232 buffer fills up rapidly (at around 0.8 ms).

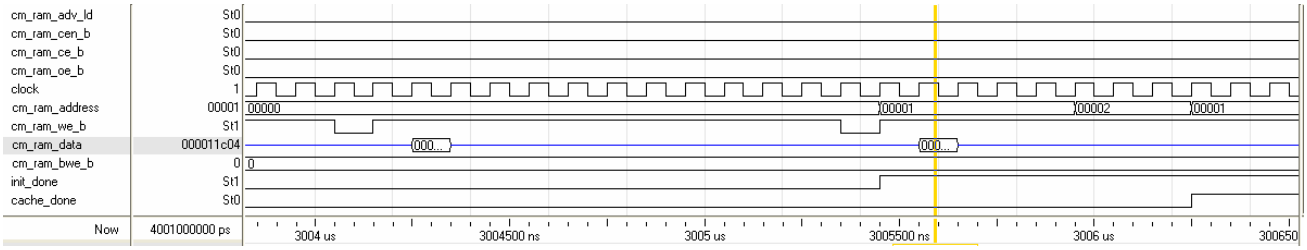

Figure 5 – Reception, storage, and caching of a (fake) 2 byte-long class file by the Class Memory Manager. This is the old way methods were cached when the J Computer only had the ability to execute single methods.

B Supported Bytecodes

The list of supported bytecodes is not included in this version of the report.

The following bytecodes which are not used in Java ME CLDC v1.0a are used to implement specific native features of the J Computer. Values in square brackets indicate immediate arguments in the bytecode stream.

Native bytecodes					**Description**
39	**0x27**	**x_rpc_invoke**			Invoke [method index]
40	**0x28**	**x_rpc_push**			Pop top byte, push to RPC bus
41	**0x29**	**x_rpc_wait_for_ret**			Wait for RPC return, push ret value
142	**0x8E**	**x_pop_rs232**			Pop top byte to RS-232 send FIFO
174	**0xAE**	**x_newframe**			New frame [params,localvars]

C J Computer Archive (.jca) File Format

```
<-- lower index byte in file, each item represents one byte
filesize[3]  filesize[2]  filesize[1]  filesize[0]
(does not count 4 bytes for file size)
numconsts[1]     numconsts[0]
numfields[1]     numfields[0]
nummethods[1]    nummethods[0]
numother[1]      numother[0]
const[0][1]  const[0][0]  const[0][3]  const[0][2]
(other consts)
field[0][1]  field[0][0]  field[0][3]  field[0][2]
(other fields)
methodaddr[0][1]    methodaddr[0][0]
methodaddr[1][1]    methodaddr[1][0]
methodaddr[2][1]    methodaddr[2][0]
methodaddr[3][1]    methodaddr[3][0]
(other method addresses, number of methods must be multiple of 4)
8'hFF (marker byte between metadata and code)
methodsize[0][0]    methodsize[0][1]    methodsize[0][2]    methodsize[0][3]
(does not count 4 bytes for method size)
methodcode[0][0]
methodcode[0][1]
methodcode[0][2]
(rest of method code)
methodsize[1][0]    methodsize[1][1]    methodsize[1][2]    methodsize[1][3]
(does not count 4 bytes for method size)
methodcode[1][0]
methodcode[1][1]
methodcode[1][2]
(rest of method code)
(rest of methods)
```

D Verilog Source Code

This appendix is not included in this version of the report.