JVM Byteocode Specification is a very comprehensive document defining a Java Real Machine.

Sun had tried to build a microJava chip, without success report.

JVM hid a Real Forth Machine. I want JavaForthMachine to prove the concept.

I used 70 JVM Bytecode to build a 32-bit integer Forth Machine.

I need 6 additional Bytecode to open up Java return stack frames.

Bytecode dealing with longs, floats, and doubles will be added when resources allows.

Then, a JavaRealMachine

I did eP32 in VHDL for a reason now forgotten.

eP32 now survived on a NIOS II kit with Quartus II From Altera/Intel, and an XP2 BreviaII kit with Diamond IDE from Lattice.

Don Golding formed the Robot AI Group to build a Forth machine on ICE40pk5 low-power FPGA kit.

He demanded that all participants coding in SystemVeriolog.

I was already converting ep32 to a Bytecode Machine on Quaruts II and Stratix II.

I need a Stratix II class FPGA to host the JavaForthMachine.

I need the Megacore library to implement multiply, divide and barrel shift to satisfy Java bytecode idiv, ishl, ishr, and iushr.

The tiny FPGA core in ICE40 cannot take in these big library components.

I finished synthesizing and simulating JavaForth on Quartus II.

I was still in VHDL. It is time to move to Verilog.

I hope to move the Outer Interpreter to jeforth615 so that people will not bother with F#.

The new metacompiler will run in HTML to generate a eJ32k.mif memory file, which will be the testbench for JavaForthMachine.

A text file is stored in an Input Buffer.

Get pulls out a byte at an autoincreamenting InPtr, and moves the byte to the end of the word buffer until a delimiter is encountered.

This is PARSE in Outer Interpreter.

Outer Interpreter ¡¥put¡¦s bytes into the OutBuf at autoincrementing OutPtr.

Output bytes can be viewed in FPGA simulator for debugging.

Th

is is the best testbench if the simulator allows you to see these bytes.

It is now greatly simplified to:

: QUIT [ BEGIN PARSE EVALUATE AGAIN

PARSE leaves an IDIOM in the word buffer above the dictionary.

EVALUATE evaluates IDIOM.

: EVALUATE ( IDIOM IN word buffer )

NAME?

If compi @ IF compile ELSE execute THEN

ELSE NUMBER?

IF compi @ IF literal THEN

ELSE QUIT THEN

THEN .OK ;