



eForth in C

Chen-Hanson Ting

SVFIG

July 22, 2017



eForth C Implementation

- **2009**
 - **eForth_10.c** 64 byte code, gcc/cygwin
 - **eForth_11.c** 32 byte code, gcc/cygwin
- **2011**
 - **ceForth328** Atmega328P, Arduino
- **2016**
 - **espForth_43** ESP8266, Arduino
- **2017**
 - **ceForth_20** New stacks, Visual Studio
 - **cdForth_23** Byte code Sequencer



ceForth Updates

- **Moved to Visual Studio 2017**
- **Stacks changed to 256 cell circular buffers**
- **Finite State Machine replaced by Byte Code Sequencer**
- **General cleanup**
- **New manual**



Virtual Forth Machine

- **67 pseudo instructions or byte code**
- **Byte Code Sequencer executes byte code**
- **Primitive commands have byte code**
- **Compound commands have token lists**



New Stacks

- **256 cell circular buffers**
- **1 byte stack pointers**
- **No underflow, no overflow**

```
long rack[256] = { 0 };
```

```
long stack[256] = { 0 };
```

```
unsigned char R, S;
```

```
long top = 0;
```



Finite State Machine

```
void main(void)
{ int phase=0;clock=0;P=0;IP=0;S=0;R=0;top=0;
  cData = (char*)data;
  while (TRUE) { phase = clock & 7;
    switch (phase) {
      case 0: fetch_decode(); break;
      case 1: execute(I1); break;
      case 2: execute(I2); break;
      case 3: execute(I3); break;
      case 4: execute(I4); break;
      case 5: jump(); break;
      case 6: jump(); break;
      case 7: jump(); }
    clock += 1;
  } }
```



Byte Code Sequencer

```
coid main(void) {  
    P=0 ; WP=4 ; IP=0 ; S=0 ; R=0 ; top=0 ;  
    cData=(char*) data ;  
    while (TRUE) {  
        bytecode=(char) cData [P++] ;  
        execute (bytecode) ;  
    }  
}
```



Inner Interpreter

```
void next(void)
{ P = data[IP>>2];
  WP = P + 4;
  IP += 4; }
```




Address Interpreter

```
void dolist(void)
{ rack[(char)++R] = IP;
  IP = WP;
  next(); }
void exitt(void)
{ IP = (long)rack[(char)R--];
  next(); }
```



Primitive Commands

Link Field: 32 bit pointer

Name Field: Length byte

Name string, null
filled to word
boundary

Code Field: Byte code

Terminator: next()



Compound Commands

Link Field:	32 bit pointer
Name Field:	Length byte Name string, null filled to word boundary
Code Field:	dolist()
Parameter Field:	Token list
Terminator:	EXIT



Liars Do Benchmark

: TEST

FOR 100000 FOR NEXT NEXT ;

ceForth_23

6M cycles/sec

ceForth_22

4M cycles/sec

espForth_43

0.25M cycles/sec

586eForth502

270M cycles/sec



Demos

ceForth_23

ceForth_22

espForth_43

586eForth502