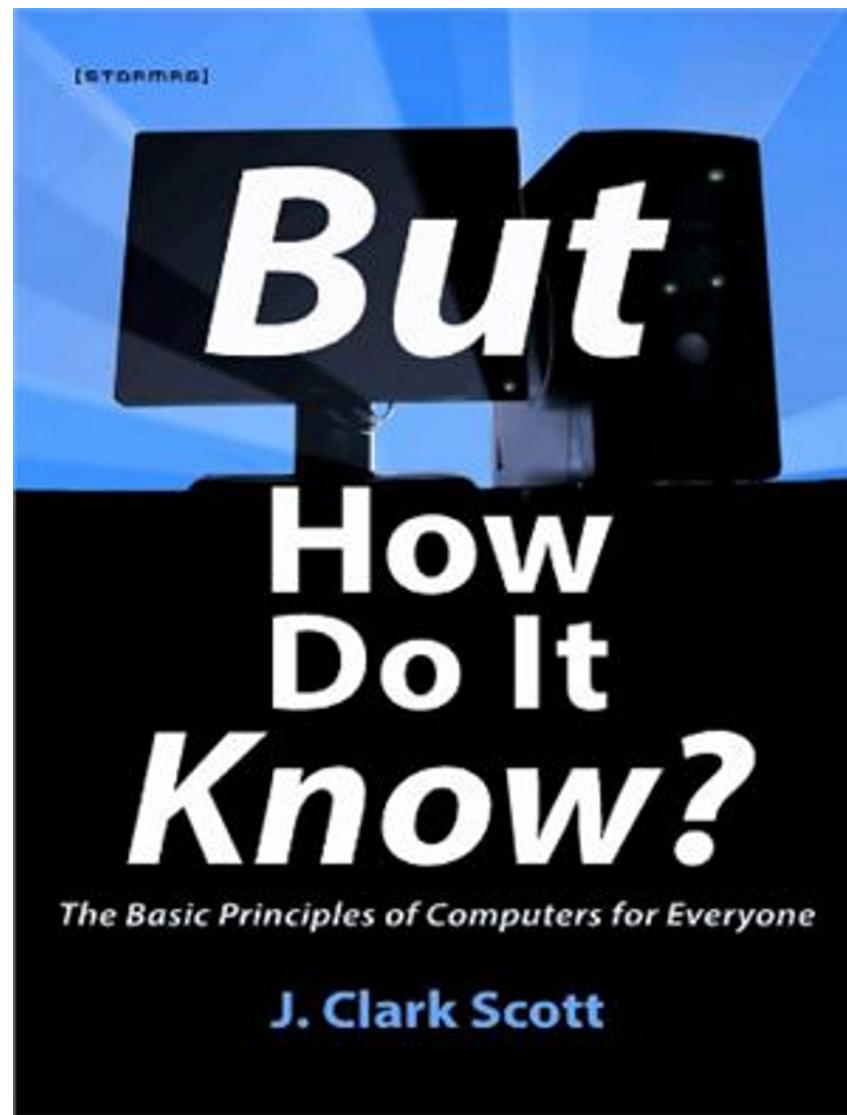


ScottE CPU

designed by Artur Faltenberg

Original Idea



Overview

- [*https://bitbucket.org/faltenberg/scotte.git*](https://bitbucket.org/faltenberg/scotte.git)
- 8bit RISC CPU
- 18 instructions
- 4 general-purpose registers
- 1KB address space (256B ROM, 256B stack, 256B RAM, 256B IO)
- performance: 2 clock ticks per instruction
- example: fibonacci(10) in 25s at 16Hz clock speed
- transistors: approx. 6000

Registers and Flags

- R0 – general-purpose
- R1 – general-purpose
- R2 – general-purpose
- R3 – general-purpose
- SP – stack pointer
- PC – program counter
- FR – flag register
 - FR[0] – zero flag
 - FR[1] – sign flag
 - FR[2] – overflow flag
 - FR[3] – carry flag
 - FR[4,5] – index flag ix,
selects active region
for ldr and str
 - FR[6] – halt flag
 - FR[7] – manual mode

Instruction Set

Instruction	Encoding	Effect
six rom stack ram io	0000 10 nn	$ix = nn$
halt	0000 11 xx	ignore clock
data ra, imm8	0001 ra xx imm[7..0]	$ra = imm$
ldr ra, [rb]	0010 ra rb	$ra = mem[(ix, rb)]$
str ra, [rb]	0011 ra rb	$mem[(ix, rb)] = ra$
jalr ra	0100 ra rb	$pc = rb; ra = pc+1$
b<cond> imm8	0101 cond imm[7..0]	$pc += cond ? imm : 1$
push ra	0110 ra xx	$stack[--sp] = ra$
pop ra	0111 ra xx	$ra = stack[sp++]$
add ra, rb	1000 ra rb	$ra = ra + rb$ (vcsz)
sub ra, rb	1001 ra rb	$ra = ra - rb$ (vcsz)
shl ra, rb	1010 ra rb	$ra = rb \ll 1$ (vcsz)
shr ra, rb	1011 ra rb	$ra = rb \gg 1$ (vcsz)
and ra, rb	1100 ra rb	$ra = ra \& rb$ (vcsz)
orr ra, rb	1101 ra rb	$ra = ra rb$ (vcsz)
xor ra, rb	1110 ra rb	$ra = ra ^ rb$ (vcsz)
cmp ra, rb	1111 ra rb	$ra - rb$ (vcsz)

Instruction Set (cont'd)

Instruction	Encoding	Effect
br imm8	0101 0000 imm[7..0]	branch always
bzs imm8	0101 0001 imm[7..0]	branch if fr[z] == 1
bss imm8	0101 0010 imm[7..0]	branch if fr[s] == 1
bov imm8	0101 0011 imm[7..0]	branch if fr[v] == 1
bcs imm8	0101 0100 imm[7..0]	branch if fr[c] == 1
beq imm8	0101 0001 imm[7..0]	branch on equal
bltu imm8	0101 0100 imm[7..0]	branch on uint <
bgeu imm8	0101 0101 imm[7..0]	branch on uint >=
blt imm8	0101 0110 imm[7..0]	branch on int <
bge imm8	0101 0111 imm[7..0]	branch on int >=

Datapath

