

MEMOCODE 2003 Panel

Stephen A. Edwards

Department of Computer Science,
Columbia University

www.cs.columbia.edu/~sedwards

sedwards@cs.columbia.edu

Q: Should the space of implementation possibilities be determined by the abilities of high-level synthesis and validation?

Q: Should the space of implementation possibilities be determined by the abilities of high-level synthesis and validation?

A: We do not have a choice if we use high-level synthesis and validation.

Euclid's Algorithm on the PDP-11

```
.globl _gcd
.text
_gcd:
    jsr r5, rsave
L2:  mov 4(r5), r1
     sxt r0
     div 6(r5), r0
     mov r1, -10(r5)
     jeq L3
     mov 6(r5), 4(r5)
     mov -10(r5), 6(r5)
     jbr L2
L3:  mov 6(r5), r0
     jbr L1
L1:  jmp rretrn
```

GPRs: r0-r7
r7=PC, r6=SP, r5=FP

Save SP in FP

r1 = n

sign extend

r0, r1 = m / n

r = r1 (m % n)

if r == 0 goto L3

m = n

n = r

r0 = n

non-optimizing compiler

return r0 (n)

Architectures for Real-Time Software

Like hardware, need to focus on worst-case performance.

Predictability at the expense of the average case.

Bad

Branch speculation

Caches

Pipeline stalls

Superscalar/OOO

Page Faults

Good

Expected/unexpected branch inst.

Scratchpad memories

Software-managed pipelines

VLIW

Software-managed MMUs

Goals

Compiler able to provide tight worst-case time bounds.

Compiler optimizations for meeting timing constraints.

Designers able to estimate software timing.

Designers able to write time-critical software.