The J1 CPU

James Bowman
Willow Garage, Inc
http://www.willowgarage.com

November 19, 2010

The PR2

Six cameras, all connected via Ethernet:

- Two wide-angle color cameras in head
- Two narrow-angle monochrome cameras in head
- One color camera in each hand

Nobody makes a small enough Ethernet camera, so we made one

The WGE

Ingredients:

- Aptina 640x480 sensor
- 100BASE-TX Ethernet PHY
- 8 Mbit flash
- Xilinx Spartan®FPGA - XC3S500E

Requirements:

- uncompressed 640x480 at 30 fps, is 9 Mbytes/s over UDP
- external sync (<30 μs) for stereo
- field software upgrade

Constraints:

- 36 Kbytes of RAM in FPGA, 16 KBytes for code
- Imager does not wait
- Must handle network commands while streaming video

First version

- MicroBlaze core
- gcc based toolchain from Xilinx
- 32-bit, 40 MIPS

Running out

- code took all of the 16KByte
- not fast enough to stream video data...
- ...so needed DMA controllers
- quite a lot of Verilog

Not alone

... actually implementing that code with the limited instruction space provided by the BRAM of the microblaze was a complete nightmare. Code optimization went from an afterthought to the top of our priority list, after each step we took and after every function we coded, it was necessary to optimize the code in order to fit in our extremely limited instruction space, otherwise, debugging was impossible because compilation was impossible.

`http://www.cs.columbia.edu/~sedwards/classes/2006/4840/reports/Web-Server.pdf`

J1: minimal CPU core

- 16-bit instructions, data
- 100MIPS fast enough to stream video data in software
- unencoded hardwired instruction encoding

Instruction formats

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	value															literal
0	0	0	target													jump
0	0	1	target													conditional jump
0	1	0	target													call
0	1	1	$R \rightarrow PC$	T'				$T \rightarrow N$	$T \rightarrow R$	$N \rightarrow [T]$		rstack ±		dstack ±		ALU

ALU operations

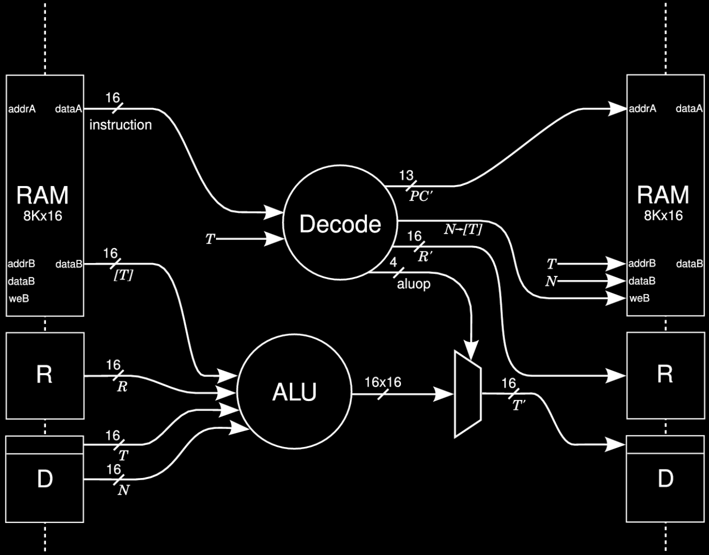
code	operation
0	T
1	N
2	$T + N$
3	$T \mathtt{and} N$
4	$T \mathtt{or} N$
5	$T \mathtt{xor} N$
6	$\sim T$
7	$N = T$
8	$N < T$
9	$N \mathtt{rshift} T$
10	$T - 1$
11	R
12	$[T]$
13	$N \mathtt{lshift} T$
14	$\mathtt{depth}$
15	$N \mathtt{u<} T$

Max Factoring

```
: 1+        1 + ;
: negate    invert 1+ ;
: -         negate + ;
```

- call is always single-cycle
- returns are free
- deep stacks

Cycle timing



Code Size

component	MicroBlaze	J1
	code size (bytes)	
I^2C	948	132
SPI	180	104
flash	948	316
ARP responder	500	122
entire program	16380	6349

Comparisons

	MicroBlaze+C	J1+Forth
CPU lines of Verilog	6000	200
program (Kbytes)	16	6
Clock (MHz)	42	80
Toolchain build time	??	about zero

Conclusion

- Much easier to make a Forth CPU
- Much less code so more maintainable
- We can fit more features in