A **hol** Theory of n [illegible] t [illegible]ords

Anthhny Fh

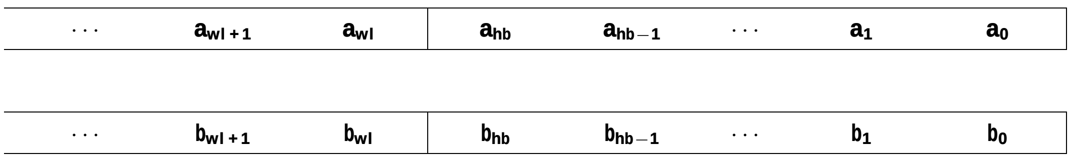
1 Introduction

This work forms part of an epsrc funded project on the formal speci cation and veri cation of the A M6 microprocessor. Under this project, Graham Birtwistle's group in Leeds have produced detailed ml speci cations of the A M instruction set and of the A M6. At Cambridge, the hol theorem prover has been used to verify a design that is closely based on the A M6. In order to model the A M instruction set and the processor implementation in hol, a model of 32-bit words was required. Moreover, the intention was

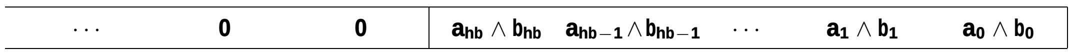
This set of equations does satisfy some of the properties of n

used to partition an algebra A's carrier set into classes

$$\cdots \quad \mathbf{b}_{h+2} \; \mathbf{b}_{h+1} \quad \mathbf{b}_{h} \quad \mathbf{b}_{h-1} \quad \cdots \quad \mathbf{b}_{l+1} \quad \mathbf{b}_{l} \quad \mathbf{b}_{l-1} \quad \cdots \quad \mathbf{b}_{1}$$



(a) The operands a and b.



(b) The whrd a and b = bitwise wl ∧ a b.

Figure 3: Bitwise Conjunction.

The function bitwise : N ! (B ! B ! B) ! N ! N ! N is used to de ne logical operations. It has a (primitive) recursive de nition:

bitwise 0 R a b = 0

bitwise (n + 1) R a b = bitwise n R a b + sbit ((bit n a) R Td(j)Tj [illegible]

5 Constructing a Word Algebra

is a well-de ned bijection betw w

Algebra: Bits

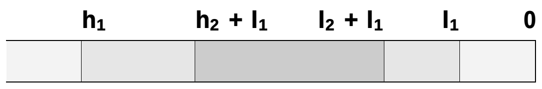
Carrier Sets

$\mathbb{B}; \mathbb{N}$

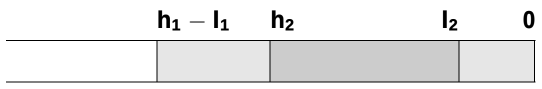
Constants

true; **false** $\in \mathbb{B}$

0;



(a) Natural number n, split into bit regions.



(b) n' = bits

6.2.2 Shifting

Shifting by zero and shifting the word 0 is an identity mapping.

Theorem 9.

$$\mathrm{lsl}\ 0\ w = \mathrm{lsr}\ 0\ w = \mathrm{asr}\ 0\ w = \mathrm{ror}\ 0\ w = w$$

and

$$\mathrm{lsl}\ x\ 0 = \mathrm{lsr}\ x\ 0 = \mathrm{asr}\ x\ 0 = \mathrm{ror}\ x\ 0 = 0:$$

The arithmetic and rotate right shifts are also identity maps forthe word T.

Theorem 10. $\mathrm{asr}\ x\ T = \mathrm{ror}\ x\ T = T.$

Theorem 11. Shifting is additive:

$$\begin{aligned}\mathrm{lsl}\ (m + n)\ w &= \mathrm{lsl}\ n\ (\mathrm{lsl}\ m\ w)\\ \mathrm{lsr}\ (m + n)\ w &= \mathrm{lsr}\ n\ (\mathrm{lsr}\ m\ w)\\ \mathrm{asr}\ (m + n)\ w &= \mathrm{asr}\ n\ (\mathrm{asr}\ m\ w)\\ \mathrm{ror}\ (m + n)\ w &= \mathrm{ror}\ n\ (\mathrm{ror}\ m\ w):\end{aligned}$$

Logical left and right shifts reach a xed point at wl app14 0 Tdate()Tj /R151 10.9091 Tf 4.23wions,0 Td 3.

7.1 The Word Type

Equivalence types can be declared in hol using the function define_equivalence_type **from the** equivType **package. Creating an equivalence type is analogous to asserting the existence of the set A= .**

7.3 Mappings to and from the Natural Numbers

Natural numbers can be mapped to words with the function `n2w:num  word32`**, de ned by**

```
`def n2w a = mk_word32 ($== a)
```

The term `$== n` **represents the equivalence class of** `n`**; it is a map characterising the set of all numbers** , de n

8 Conclusion

A hol theory of n-bit words has been presented with reference to the quotient

[10] Warren A. Hunt, Jr.