On Size 2 Rush Hour Logic

John Tromp

December 18, 2000

Abstract

In [1] Rush Hour Logic is shown to support polynomial space computation. The auhors consider cars of size 2 and 3, saying that the latter are crucial to their construction, and conjecture that cars of size 2 only, which we'll call *Size 2 Rush Hour Logic*, do not support polynomial space computation. We settle this question by showing that all required building blocks are constructible in Size 2 Rush Hour Logic.

1 Introduction

We show signalling cars in white, and anchoring cars in black and gray. Any black part of a car indicates a cell which is necessarily always occupied.

2 The intersection

The block in Figure 1 allows the signals from bottom to top and from right to left to pass without harm.

3 The or

An or $c = a \vee b$ must allow all triples $(a, b, c) \in \{0, 1\}^3$ as states except $(0, 0, 1)$. Furthermore, it must allow all transitions between allowed states that differ in only one bit. We construct such a general or in terms of two weaker ors, or' and or". Or' has essentially two versions of state $(1, 1, 1)$ and from each one disallows a transition to either $(1, 0, 1)$ or $(0, 1, 1)$. Or" instead disallows the transition from $(0, 1, 0)$ to $(0, 1, 1)$. A general or may be computed as

$$a \vee b = (a \vee' b) \vee'' (a \vee' b).$$

The block in Figure 2 computes an or' of the left and bottom inputs and or"s the result with the right input to yield the output on top. Note that or" is like the inverse of or', in the sense of swapping the roles of one input with the output.

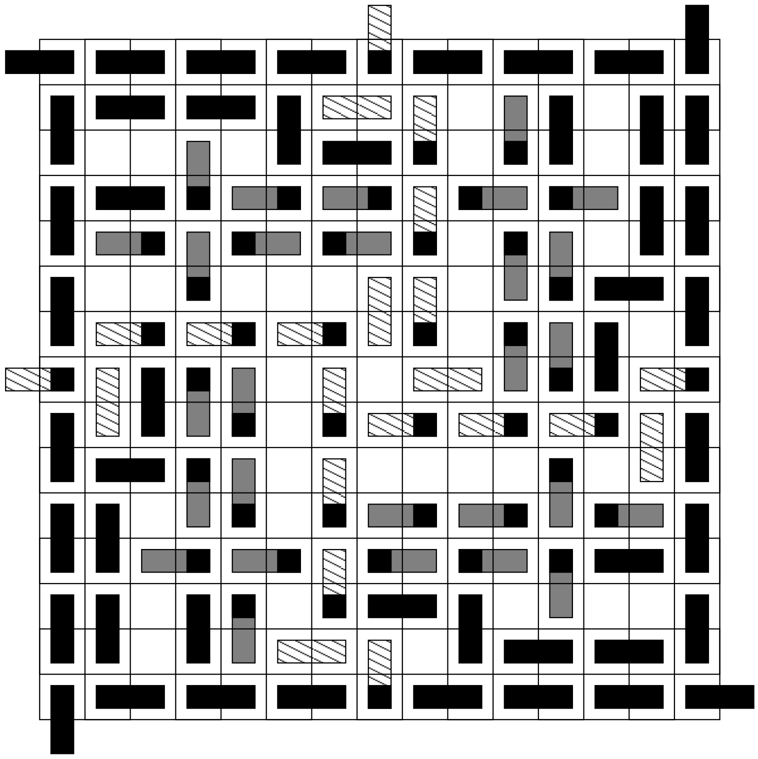

Figure 1: Size 2 Rush Hour Logic intersection block.

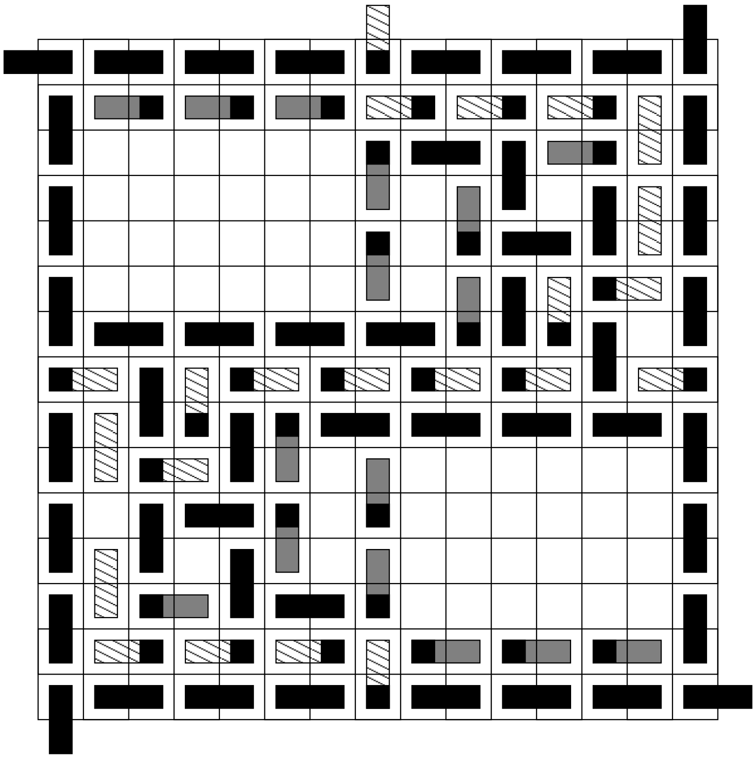

Figure 2: Size 2 Rush Hour Logic or block.

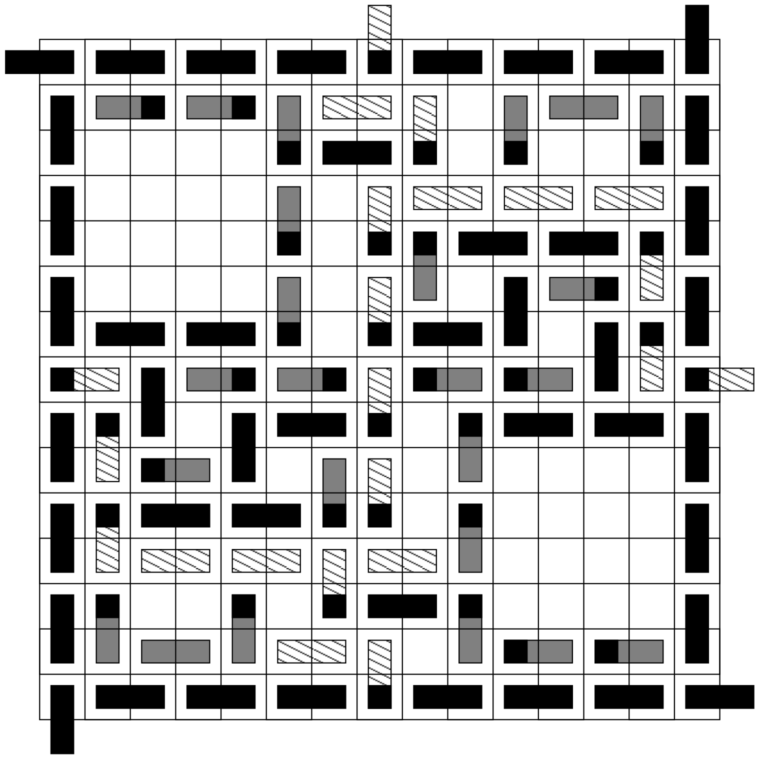

Figure 3: Size 2 Rush Hour Logic and/split block.

4 The and/split

The block in Figure 3 computes the logical and of the left and bottom inputs and splits the result to the top and right outputs. Note that split is the inverse of and.

5 Acknowledgements

References

[1] Gary. W. Flake and Eric. B. Baum, Rush Hour is PSPACE-complete, submitted to *Theoretical Computer Science.*