Reviewer: 1  
  
Comments to the Author  
The paper describes a framework to verify the SparkV8 assembly code. The assembly code is used in some OS kernel implementation to access hardware and to improve efficiency. People have proposed an approach to verify OS kernel code, in which assembly code is replaced with logic formulae expressing their semantics. The author formalized the behaviors of the SparkV8 code in this paper, which complements the previous approach under a refinement framework. That is, people can first show the kernel code is correct with assembly code replaced with logic formulae describing its semantics, then show a refinement relation between the logic formula and the concrete semantics provided in this paper.  
  
In general, I think this is a nice work with significant practical impact and should be accepted for publication.   
  
Reviewer: 2  
  
Comments to the Author  
Compared to the original version, the main additions in the current revision are proofs of soundness of the logic for refinement verification, and more detailed proofs using the logic in the context switch routine case study. These additions significantly strengthened the paper. I am now satisfied with the current version, and recommend it to be accepted. Below are some possible minor typos:  
  
Page 2, right, line 13: ChangY -> ChangeY.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Thanks, fixed!

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Page 22, left, line 7: parts of the formula are highlighted in yellow without explanation.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Thanks, we have supplemented some explanations about the formula highlighted in yellow. Please find it in colored in blue.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Page 29, left, line 5: proof effect -> proof effort?

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Thanks, fixed!

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
Reviewer: 3  
  
Comments to the Author  
This review is for a revised version of an earlier submission. The authors addressed many of the comments from an earlier review, and especially extended the proofs and proof sketches in the paper. The case study of a context switch routine was extended with a proof overview and now illustrates how the approach is applied. Overall, these improvements add considerably rigorousness to the paper, and I only have some minor comments below.  
  
The syntax text explanations could be further improved with explicit references or closer notational similarity to the syntax definitions in the figures. For example, Fig. 3 and text on p. 5: could say "The operand expression (OpExp)..., and address expression (AddExp)" etc.   
  
The proof sketch of Lemma 2 still needs a bit more explanation (how is Lemma 1 applied here?).  
  
The global program transition rule on p. 17 should mention M' in the conclusion.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Thanks, it’s a typo and we have fixed M to M’.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
Minor comments  
--------------  
1. p. 2: ChangY --> ChangeY  
2. p. 4: implementated --> implemented  
3. p. 5: Simple instruction "nop" is not mentioned in the text  
4. p. 7: state of processor, and there is --> state of the processor, and there it is  
5. p. 8: we use block-based model --> we use a block-based model  
6. p. 8: safe and restore instruction rotate --> save and restore instructions rotate  
7. p. 10 (Section 3 overview): mention that refinement is added in Sect. 4  
8. p. 11: assign a \theta --> assign a specification \theta  
9. p. 14: shown as the left side of Fig. 12 -->  shown on the left side of Fig. 12  
10. p. 16: two parts : --> two parts:  
11. p. 16: in specific form as mentioned before --> in the specific form mentioned before  
12. p. 16: states [...] is defined --> states [...] are defined  
13. p. 16: consts: --> consists of:  
14. p. 16: omit special register --> omit special registers  
15. p. 16: program, We --> program, we  
16. p. 16: of switch primitive --> of the switch primitive  
17. p. 16: of current thread --> of the current thread  
18. p. 21: primtive --> primitive  
19. p. 21: abtract --> abstract  
20. p. 21: it's execution --> its execution  
21. p. 24: exisits --> exists  
22. p. 26: It saves local and in registers of current window --> It saves the local and in registers of the current window  
23. p. 26: and call reg\_save --> and calls reg\_save  
24. p. 27: Usedwindow --> UsedWindows (?)  
25. p. 27: of context switch routine --> of the context switch routine  
26. p. 27: set of abstract assembly primitive --> [...] primitives  
27. p. 27: each code blocks --> each code block  
28. p. 28: of context switch/of thread pool --> of the ...  
29. p. 30: implementated --> implemented