theory *SALMemPlatform* = *SALOverflowPlatform*:

1 SAL Memory Platform

— This platform accepts programs that are types safe, do not cause arithmetics overflows and do not use memory addresses outside [0, MAXMEM].

1.1 Platform Definition

constdefs
MAXMEM :: *nat*
MAXMEM ≡ *20*

constdefs
safeFm :: *SALprogram* ⇒ *pos* ⇒ *SALform*
safeFm prg pc ≡
(*let* (*pn*,*i*) = *pc in*
(*case* (*cmd prg pc*)
of None ⇒ *FalseF*
| *Some ins* ⇒
(*case ins*
of SET x n ⇒ (λ (*p*, *m*, *e*). *x* < *MAXMEM*)
| *ADD x y* ⇒ (λ (*p*, *m*, *e*). *x* < *MAXMEM* ∧ *y* < *MAXMEM*)
| *SUB x y* ⇒ (λ (*p*, *m*, *e*). *x* < *MAXMEM* ∧ *y* < *MAXMEM*)
| *INC x* ⇒ (λ (*p*, *m*, *e*). *x* < *MAXMEM*)
| *JMPEQ x y t* ⇒ (λ (*p*, *m*, *e*). *x* < *MAXMEM* ∧ *y* < *MAXMEM*)
| *JMPL x y t* ⇒ (λ (*p*, *m*, *e*). *x* < *MAXMEM* ∧ *y* < *MAXMEM*)
| *JLE x y t* ⇒ (λ (*p*, *m*, *e*). *x* < *MAXMEM* ∧ *y* < *MAXMEM*)
| *JMPB t* ⇒ λ (*p*,*m*,*e*). *True*
| *CALL x pn′* ⇒ (λ (*p*, *m*, *e*). *x* < *MAXMEM*)
| *RET x* ⇒ (λ (*p*, *m*, *e*). *x* < *MAXMEM*)
| *MOV s t* ⇒ (λ (*p*, *m*, *e*). (∀ *ns*. *m s* = *NAT ns* ⟶ *ns* < *MAXMEM*) ∧
(∀ *nt*. *m t* = *NAT nt* ⟶ *nt* < *MAXMEM*))
| *HALT* ⇒ λ (*p*,*m*,*e*). *True*
)
)
)

constdefs
safeF:: *SALprogram* ⇒ *pos* ⇒ *SALform*
safeF prg p ≡ *Conj* [(*SALSafetyLogic.safeF prg p*),*safeFm prg p*]

lemma[*code*]:
safeF prg p ≡ *let a*=(*SALSafetyLogic.safeF prg p*); *b*=(*safeFm prg p*)

in (*term* (*Conj* [(*to-term a*),(*to-term b*)]))
apply (*simp only*: *safeF-def to-term-def term-def Let-def*)
done

constdefs
vcgSALm :: *SALprogram* ⇒ *SALform*
vcgSALm prg ≡ *vcG Conj Impl FalseF ipc initF safeF succsF wpF domC domA anF prg*

generate-code (*vcgSALT.ML*) [*term-of*]
vcg = *vcgSALT*

constdefs
isafeF::*SALprogram* ⇒ *pos* ⇒ *SALform*
isafeF prg pc ≡ *isafe*(*domC prg*,*prg*,*anF prg*,*pc*,*FalseF*,*Conj*,*Impl*,*safeF*,*succsF*,*wpF*)

constdefs
isafeP::*SALprogram* ⇒ *SALstate set* (*isafe*$_{\square}$- [*70*])
isafeP prg ≡ (*isafeP'* *effS valid initF isafeF prg*)

lemma *isafeP-induct*:
⟦ *s* ∈ (*isafeP prg*);
⋀ *s*. ⟦ *valid prg s* (*initF prg*) ⟧ ⟹ *P s*;
⋀ *s s'*. ⟦ *s* ∈ (*isafeP prg*); *valid prg s* (*isafeF prg* (*fst s*)); *valid prg s'* (*isafeF prg* (*fst s'*)); (*s*,*s'*) ∈ (*effS prg*); *P s* ⟧ ⟹ *P s'* ⟧ ⟹ *P s***done**

constdefs
provable :: *SALprogram* ⇒ *SALform* ⇒ *bool* ((- ⊢ -) [*61*,*60*] *60*)
provable prg f ≡ ∀ *s*. *s* ∈ (*isafeP prg*) ⟶ *valid prg s f*
end