theory *SALMemFWInst* = *SALMemPlatform* + *SALOverflowFWInst*:

— Instantiating the VCG Framework

lemma *safeF-mono*:
(*safeF prg p*) *s* ⟹ (*SALSafetyLogic.safeF prg p*) *s*
by (*simp add*: *safeF-def Conj-def*)

lemma *wpF-mono*:
⟦ (*wpF prg p p′ Q*) *s*; (∀ *s*. *Q s* ⟶ *Q′ s*) ⟧ ⟹ (*wpF prg p p′ Q′*) *s***done**

lemma *isafe-mono*:
⋀ *s*. *valid prg s* (*isafe* (*domC prg*,*prg*,*anF prg*,*p*,*FalseF*,*Conj*,*Impl*,*safeF*,*succsF*,*wpF*))
⟹ *valid prg s* (*isafe* (*domC prg*,*prg*,*anF prg*,*p*,*FalseF*,*Conj*,*Impl*,*SALSafetyLogic.safeF*,*succsF*,*wpF*))
done

lemma *isafeP-mono*:
⋀ *s prg*. ⟦ *s* ∈ *isafeP prg* ⟧ ⟹ *s* ∈ (*SALSafetyLogic.isafeP prg*)**done**

theorem *SAL-VCG-Ins*:
*VerificationConditionGenerator Conj Impl TrueF FalseF valid provable effS wpF succsF initF ipc SALMemPlatform.safeF anF domC wf***done**

1 Platform Soundness

constdefs *isSafe*::*SALprogram* ⇒ *bool*
isSafe prg ≡ (∀ *s s′*. *prg*,*s* ⊨ *initF prg* ∧ (*s*,*s′*) ∈ (*effS prg*)* ⟶ *prg*,*s′* ⊨ *safeF prg* (*fst s′*))

theorem *platform-soundness*:
⟦ *wf prg*; *provable prg* (*vcgSALm prg*) ⟧ ⟹ *isSafe prg***done**

end