



CAL actor language	
<pre>actor ID () In ==> Out : action In: [a] ==> Out</pre>	a end
end	actor ID () In ==> Out :
	<pre>action [a] ==> [a] end end</pre>
actor Add () Input	1, Input2 ==> Output:
<pre>action [a], [b] end</pre>	==> [a + b] end
actor Add	dSeq () Input ==> Output:
actic	on [a, b] ==> [a + b] end
Writing actors: the CAL actor langua Jörn W. Janneck	ge Xilinx Research Labs San Jose, CA

CAL actor language parameters	
<pre>actor Scale (k) Input ==> Output: action [a] ==> [k * a] end end</pre>	parametric actor definitions represent a <i>family</i> of actors
Writing actors: the CAL actor language	Xilinx Research Labs San Jose, CA



















CAL actor language priorities (the harmless case)	
actor Route () A ==> X, Y, Z:	
action $[v] ==> X$: $[v]$ guard $P(v)$ end	
action $[v] ==> Y$: $[v]$ guard $Q(v)$ and not $P(v)$ end	
action $[v] => Z$: $[v]$	actor Route () A ==> X, Y, Z:
guard not $Q(\nabla)$ and not $P(\nabla)$ end end	toX: action [v] ==> X: [v] guard P(v) end
	toY: action [v] ==> Y: [v] guard Q(v) end
	toZ: action $[v] \implies Z: [v]$ end
	<pre>priority toX > toY > toZ;</pre>
	end end
Writing actors: the CAL actor language	Xilinx Research Labs San Jose, CA































