			\(\lambda \text{ ackWir} \) \(\lambda \text{ dataWir} \) \(\lambda \text{ dataW} \) \(\lambda \text{ outposts of ackS} \) \(\lambda \text{ sends} \) \(\lambda \text{ senderPc} = [S \) \(\lambda \text{ senderPc} = [S \) \(\text{ SynA} \) \(\text{ ReceiveBut of ackS} \)	A senderState = "WAITING" A ackWireIn = <<>> A ackWireOut = <<>> A dataWireOut = <<>> A ck SenNum = 0 A senderIdx = 1 A buffer = <<1> A senderIdx = 1 A buffer = <1> A senderIdx = 1 A senderPc = SendSynAck > "A ", ReceiveButFirst -> "A R", Sender -> "A s" A receiverState = "WAITING" A receiverState = "WAITING	A senderState = "WAITING"	\(\) \(\sender \) \(\tau \) \(\(\lambda\) da \(\lambda\) da \(\lambda\) da \(\lambda\) da \(\lambda\) ou \(\lambda\) \(\lambda\) \(\lambda\) \(\lambda\) \(\lambda\) \(\lambda\) senderPc \(\lambda\) Rece	rState = "WAITING" kWireIn = <<>> WireOut = <<>> WireOut = <<>> An = 4 = <>> putreIn = <<>> putreIn = <<>> senderIdx = 1 buffer = <<>> I SendSynack > "A_S", ACK > "A," yNack > "A," reButFirst > "A," reButFirst > "A," reButFirst > "A," reButFirst > "A," restate = "WAITING"						
	A senderState = "WAITING" A ackWrein = <<>> A ackWreiotut = <<>> A dataWreiotut = <<>> A dataWreiotut = <<>> A dataWreiotut = <<>> A output = <<>> A output = <<>> A output = < A senderIdx = 2 A buffer = <<>> A buffer = <<>> A buffer = <<>> A ck -> "A" Synack -> "A" Synack -> "A" Sondor -> "A " Sondor -> "A " Sondor -> "A " NeceiverState = "WAITING"			\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireIn} = <<>> \\ \lambda \text{ackWireOut} = <<>> \\ \lambda \text{dataWireOut} = <<>> \\ \lambda \text{ackSeqNum} = 0 \\ \lambda \text{ackSeqNum} = 0 \\ \lambda \text{senderIdx} = 3 \\ \lambda \text{buffer} = <<>> \\ \lambda \text{senderJdx} = 3 \\ \lambda \text{buffer} = <<>> \\ \lambda \text{senderPc} = [SendSynAck -> \text{"A_S"}, \\ \text{ACK} -> \text{"A}, \\ \text{SynAck} -> \text{"A}, \\ \text{Sender} -> \text{"A} \text{"A}, \\ \text{Sender} -> \text{"A} \text{S"} \] \\ \lambda \text{receiverState} = \text{"WAITING"}			A senderState = "WAITING"							
	/\ senderState = "WAITING"	\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireIn} = \lambda \lambda \\ \(\lambda \text{ackWireOut} = \lambda \lambda \\ \(\lambda \text{dataWireOut} = \lambda \lambda \\ \(\lambda \text{dataWireIn} = \lambda \lambda \\ \(\lambda \text{ackSeqNum} = 0 \\ \(\lambda \text{senderIdx} = 2 \\ \(\lambda \text{buffer} = \lambda \lambda \lambda \text{senderIdx} = 2 \\ \(\lambda \text{buffer} = \lambda \lambda \lambda \lambda \lambda \lambda \text{senderJake} \rangle \lambda	\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireOut} = <<>> \\ \lambda \text{dataWireOut} = <<>> \\ \lambda \text{dataWireIn} = <<>> \\ \lambda \text{output} = <<1>> \\ \lambda \text{ackSeqNum} = 0 \\ \lambda \text{senderIdx} = 2 \\ \lambda \text{buffer} = <<2>> \\ \lambda \text{senderPc} = [\text{SendSynAck} -> \text{"A_S"}, \\ \text{ACK} -> \text{"A_"}, \\ \text{Sender} \text{PceiveButFirst} -> \text{"A_R"}, \\ \text{Sender} -> \text{"A_S"}] \\ \lambda \text{receiverState} = \text{"WAITING"}	\(\text{senderState} = \text{"WAITING"} \\ \text{ ackWireIn} = \leqstarrow \rightarrow \\ \text{ ackWireOut} = \leqstarrow \rightarrow \\ \text{ dataWireIn} = \leqstarrow \rightarrow \\ \text{ n = 4} \\ \text{ dataWireIn} = \leqstarrow \rightarrow \\ \text{ output} = \leqstarrow \leqstarrow \text{ ackSeqNum} = 0 \\ \text{ \choose \text{ senderIdx} = 3} \\ \text{ \choose \text{ buffer} = \leqstarrow \rightarrow \\ \text{ senderPc} = [SendSynAck -> \text{ "A_S"}, \\ \text{ ACK} -> \text{ "A_"}, \\ \text{ SynAck} -> \text{ "A_"}, \\ \text{ ReceiveButFirst} -> \text{ "A_R"}, \\ \text{ Sender} -> \text{ "A_S"}] \\ \text{ \text{ receiverState} = \text{ "WAITING"}}	\(\)\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	\(\text{\senderState} = \text{\"WAITINO"} \\ \text{\ackWireIn} = <<>> \\ \text{\ackWireOut} = <<>> \\ \text{\ackWireOut} = <<>> \\ \text{\ackWireIn} = <<>> \\ \text{\ackVireIn} = <<1, 2>> \\ \text{\ackSeqNum} = 0 \\ \text{\ackSeqNum} = 0 \\ \text{\senderIdx} = 3 \\ \text{\buffer} = <<3>> \\ \text{\senderPc} = [SendSynAck] -> \\ \text{\ack ACK} -> \text{\ack ACK}	"A_S",	\(\lambda \text{senderState} = \text{"WAITING"} \\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireIn} = \lambda \lambda \\ \lambda \text{ackWireOut} = \lambda \lambda \\ \lambda \text{dataWireIn} = \lambda \lambda \\ \lambda \text{output} = \lambda \lambda \lambda \\ \lambda \text{output} = \lambda \lambda \lambda \\ \lambda \text{output} = \lambda \lambda \lambda \\ \lambda \text{senderIdx} = 4 \\ \lambda \text{buffer} = \lambda \lambda \\ \lambda \text{senderIdx} = 4 \\ \lambda \text{buffer} = \lambda \lambda \\ \lambda \text{senderJdx} \rangle \lambda \lambda \\ \lambda \text{senderPc} = \left[\text{SendSynAck} \rangle -> \text{"A_S",} \\ \text{Sender} \rangle - \text{"A_",} \\ \text{Sender} \rangle - \text{"A_S",} \\ \text{Sender} \rangle - \text{"A_S",} \\ \text{N receiverState} = \text{"WAITING"}		\(\text{senderState} = \text{"WAITING"} \\ \text{ ackWireIn} = <<>> \\ \text{ ackWireOut} = <<>> \\ \text{ dataWireOut} = <<>> \\ \text{ n = 7} \\ \text{ dataWireIn} = <<>> \\ \text{ output} = <<1, 2, 3>> \\ \text{ ackSeqNum} = 0 \\ \text{ \text{ senderIdx} = 4} \\ \text{ \text{ buffer}} = <<>> \\ \text{ \text{ senderPc} = [SendSynAck -> \text{ "A_S",} \\ \text{ ACK } -> \text{ "A_",} \\ \text{ Sender } -> \text{ "A_S",} \\ \text{ Sender } -> \text{ "A_S",} \\ \text{ \text{ receiveButFirst}} -> \text{ "A_S",} \\ \text{ \text{ receiverState}} = \text{ "WAITING"}			
	\(\text{senderState} = \text{"WAITING"} \\ \text{ ackWireIn} = <<>> \\ \text{ ackWireOut} = <<>> \\ \text{ dataWireOut} = <<>> \\ \text{ dataWireIn} = <<>> \\ \text{ output} = <<1>> \\ \text{ output} = <<1>> \\ \text{ ackSeqNum} = 0 \\ \text{ \text{ senderIdx} = 3} \\ \text{ \text{ buffer} = <<>>} \\ \text{ \text{ senderPc} = [SendSynAck -> \text{ "A_S",} \\ \text{ ACK -> \text{ "A_",} \\ \text{ SynAck -> \text{ "A_ ",} \\ \text{ ReceiveButFirst -> \text{ "A_ R",} \\ \text{ Sender -> \text{ "A_ S"]} \\ \text{ \text{ receiverState} = \text{ "WAITING"} \end{array}	\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireOut} = <<>> \\ \lambda \text{ataWireOut} = <<>> \\ \lambda \text{ataWireIn} = <<>> \\ \lambda \text{ataWireIn} = <<>> \\ \lambda \text{output} = <<1, 2, 3>> \\ \lambda \text{ackSeqNum} = 0 \\ \lambda \text{senderIdx} = 2 \\ \lambda \text{buffer} = <<>> \\ \lambda \text{senderPc} = [SendSynAck -> \text{"A} \], \\ \text{SenderPc} = [SendSynAck -> \text{"A} \text{",} \\ \text{SynAck} -> \text{"A} \text{",} \\ \text{Sender} -> \text{"A} \text{",} \\ \text{Sender} -> \text{"A} \text{S"} \] \\ \lambda \text{receiverState} = \text{"WAITING"}			\(\text{senderState} = \text{"WAITING"} \\ \text{ ackWireIn} = \leq \leq \rightarrow \\ \text{ ackWireOut} = \leq \leq \rightarrow \\ \text{ \text{ dataWireOut}} = \leq \leq \rightarrow \\ \text{ \text{ n} = 0} \\ \text{ \text{ dataWireIn}} = \leq \leq \rightarrow \\ \text{ \text{ output}} = \leq \leq 1, 2 > \rightarrow \text{ \text{ ackSeqNum}} = 0 \\ \text{ \text{ senderIdx}} = 5 \\ \text{ \text{ buffer}} = \leq \leq \rightarrow \\ \text{ senderPc} = [SendSynAck -> \text{ "A_S",} \\ \text{ ACK } -> \text{ "A_",} \\ \text{ SynAck } -> \text{ "A_R",} \\ \text{ Sender } -> \text{ "A_S"} \] \\ \text{ receiverState} = \text{ "WAITING"}	\(\begin{align*} \lambda \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	tte = "WAITING" reIn = <<>> eOut = <<>> reOut = <<>> n = 6 reIn = <<>> eqNum = 0 derIdx = 3 er = <<>> endSynAck -> "A_S", K -> "A", ck -> "A_", ttFirst -> "A_R", er -> "A_S"] ate = "WAITING"	\(\lambda \text{senderState} = \text{"WAITING"} \\ \ \ \ \ \text{ackWireIn} = \lambda \lambda \rights \\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireIn} = \lambda \lambda \\ \text{\lambda tataWireOut} = \lambda \lambda \\ \text{\lambda tataWireIn} = \lambda \lambda \\ \text{\lambda tataWireIn} = \lambda \lambda \\ \text{\lambda tataWireIn} = \lambda \lambda \\ \text{\lambda totataWireIn} = \lambda \lambda \\ \text{\lambda totataWireIn} = \lambda \lambda \\ \text{\lambda totataWireIn} = \lambda \lambda \\ \text{\lambda tataWireIn} = \lambda \lambda \\ \text{\lambda totataWireIn} = 0 \text{\lambda senderIdx} = 5 \text{\lambda buffer = <<>>} \text{\lambda senderIdx} = 5 \text{\lambda buffer = <<>>} \text{\lambda senderIdx} \text{\lambda -> "A_S",} \text{\lambda SenderPc} = [SendSynAck -> "A_R", \text{\lambda Sender} -> "A_R", \text{\lambda Sender} -> "A_R", \text{\lambda Sender} -> "A_S"] \text{\lambda receiverState} = "\text{\text{\text{\lambda}ITING"}}			/\ senderState = "WAITING"		
	\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireIn} = <<>> \\ \lambda \text{ackWireOut} = <<>> \\ \lambda \text{ataWireOut} = <<>> \\ \lambda \text{ataWireIn} = <<>> \\ \lambda \text{output} = <<1>> \\ \lambda \text{ackSeqNum} = 0 \\ \lambda \text{senderIdx} = 3 \\ \lambda \text{buffer} = <<>> \\ \lambda \text{senderPc} = [\text{SendSynAck} -> \text{"A_S"}, \\ \text{ACK} -> \text{"A}, \\ \text{SynAck} -> \text{"A_T"}, \\ \text{ReceiveButFirst} -> \text{"A_R"}, \\ \text{Sender} -> \text{"A_S"}] \\ \lambda \text{receiverState} = \text{"WAITING"}	\(\) \(\sender \) \(\sender \) \(\ack \) \(\sender \) \(\ack	\(\) \(\text{senderState} = \) \"WAITING" \(\) \(\text{ackWireIn} = <<>> \) \(\) \(\text{dataWireOut} = <<>> \) \(\text{dataWireOut} = <<>> \) \(\text{dataWireIn} = <<>> \) \(\text{doutput} = <<1, 2>> \) \(\text{dataSeqNum} = 0 \) \(\text{dataSeqNum} = 0 \) \(\text{dataFerror} \) \(\text{detaRed SeqNum} = 0 \) \(\text{data SenderIdx} = 4 \) \(\text{douffer} = <<>> \) \(\text{detaRed SendSynAck} -> \"A_S", \) \(\text{ACK} -> \"A_", \) \(\text{SynAck} -> \"A_", \) \(\text{Sender} -> \"A_s" \) \(\text{ReceiveButFirst} -> \"A_s" \) \(\text{detaRed NITING} \) \(\text{detaRed NITING} \)	\(\text{senderState} = \text{"WAITING"} \\ \text{ ackWireIn} = <<>> \\ \text{ ackWireOut} = <<>> \\ \text{ dataWireOut} = <<>> \\ \text{ dataWireIn} = <<>> \\ \text{ output} = <<1, 2>> \\ \text{ ackSeqNum} = 0 \\ \text{ ackSeqNum} = 0 \\ \text{ senderIdx} = 5 \\ \text{ buffer} = <<>> \\ \text{ senderPc} = [SendSynAck -> \text{"A_S",} \\ \text{ ACK } -> \text{"A_",} \\ \text{ SynAck } -> \text{"A_T",} \\ \text{ Sender } -> \text{"A_S",} \\	\(\)\ \senderState = "WAITING" \(\)\ \ackWireIn = <<>> \)\ \(\)\ \dataWireOut = <<>> \\ \)\ \(\)\ \dataWireIn = <<1, 2>> \\ \(\)\ \ackSeqNum = 0 \\ \(\)\ \dataself \text{seqNum} = 0 \\ \(\)\ \dataself \text{senderIdx} = 5 \\ \(\)\ \dataself \text{buffer} = <<>> \\ \lambda \text{senderPc} = [SendSynAck -> "A_S", \\ \(\)\ \dataself \text{ACK } -> "A_", \\ \(\)\ \text{Sender } -> "A_s", \\ \(\)\ \\ \text{Sender } -> "A_s" \]\ \(\)\ \\ \text{receiveButFirst} -> "A_s" \]\ \(\)\ \\ \text{receiverState} = "WAITING"	\(\text{senderState} = \text{"WAITING"} \\ \text{ackWireIn} = <<>> \\ \text{AckWireOut} = <<>> \\ \text{AdtaWireOut} = <<>> \\ \text{AdtaWireIn} = <<>> \\ \text{AdtaWireIn} = <<>> \\ \text{Aoutput} = <<1, 2>> \\ \text{AoutSeqNum} = 0 \\ \text{AsenderIdx} = 5 \\ \text{Auffer} = <<>> \\ \text{Abuffer} = <<>> \\ \text{Ack} - \text{"A",} \\ \text{SenderPc} = [\text{SendSynAck} - \text{"A",} \\ \text{SynAck} - \text{"A",} \\ \text{ReceiveButFirst} - \text{"A A",} \\ \text{ReceiveButFirst} - \text{"A A",} \\ \text{Sender} - \text{"A S"]} \\ \text{\text{receiverState}} = \text{"WAITING"}	\(\)\ \senderState = "WAITING" \(\)\ \ \ackWireIn = <<>> \(\)\ \ \ \ackWireOut = <<>> \(\)\ \(\)\ \atawireOut = <<>> \(\)\ \(\)\ \atawireIn = <<>> \(\)\ \(\)\ \atawireIn = <<1, 2, 3>> \(\)\ \(\)\ \ackSeqNum = 0 \(\)\ \(\)\ \senderIdx = 6 \(\)\ \(\)\ \begin{array}\limits \text{enderIdx} = 6 \(\)\ \\ \senderPc = [SendSynAck -> "A_S", ACK -> "A", SynAck -> "A", SynAck -> "A_", ReceiveButFirst -> "A_R", Sender -> "A_S"] \(\)\ \(\)\ \receiverState = "WAITING"	\(\lambda \text{senderState} = \text{"WAITING"} \\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	\(\text{senderState} = \text{"WAITING"} \\ \text{ \ackWireIn} = \leqstarrow \rightarrow \\ \text{\ackWireOut} = \leqstarrow \rightarrow \\ \text{\ackWireIn} = \leqstarrow \rightarrow \\ \text{\ackSeqNum} = 0 \\ \text{\ackSeqNum} = 0 \\ \text{\senderIdx} = 5 \\ \text{\buffer} = \leqstarrow \rightarrow \leqstarrow \rightarrow \ri	\(\)\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	\(\text{senderState} = \text{"WAITING"} \\ \text{ ackWireIn} = \leq \leq \rightarrow \\ \text{ ackWireOut} = \leq \leq \rightarrow \\ \text{ dataWireOut} = \leq \leq \rightarrow \\ \text{ n = 7} \\ \text{ dataWireIn} = \leq \leq \rightarrow \\ \text{ output} = \leq \leq 1, 2, 3 \rightarrow \\ \text{ output} = \leq \leq 1, 2, 3 \rightarrow \\ \text{ output} = \leq \leq 1, 2, 3 \rightarrow \\ \text{ output} = \leq \leq 1, 2, 3 \rightarrow \\ \text{ ackSeqNum} = 0 \\ \text{ \text{ output} senderIdx} = 6 \\ \text{ \text{ buffer}} = \leq \leq \rightarrow \text{ \text{ Ack -> "A_",}} \\ \text{ senderPc} = \left[\text{ SendSynAck -> "A_",} \\ \text{ Sender -> "A_",} \\ \text{ Sender -> "A_s"]} \\ \text{ \text{ receiverState}} = \text{"WAITING"}	\(\) \(\senderState = \) \(\) \(\ack \)	\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireIn} = <<>> \\ \lambda \text{ackWireOut} = <<>> \\ \lambda \text{ataWireOut} = <<>> \\ \lambda \text{ataWireIn} = <<>> \\ \lambda \text{output} = <<1, 2, 3>> \\ \lambda \text{ackSeqNum} = 0 \\ \lambda \text{senderIdx} = 7 \\ \lambda \text{buffer} = <<>> \\ \lambda \text{senderJdx} = 7 \\ \lambda \text{buffer} = <<>> \\ \lambda \text{senderJdx} = \text{Name}, \\ \text{senderPc} = [\text{SendSynAck} -> \text{"A_s"}, \\ \text{SynAck} -> \text{"A_s"}, \\ \text{Sender} -> \text{"A_s"}, \\ Sender	\(\text{senderState} = \text{"WAITING"} \\ \text{ ackWireIn} = \langle < > > \\ \text{ ackWireOut} = \langle < > > \\ \text{ dataWireOut} = \langle < > > \\ \text{ dataWireIn} = \langle < > > \\ \text{ output} = \langle (1, 2, 3) > \\ \text{ ackSeqNum} = 0 \\ \text{ \langle senderIdx} = 7 \\ \text{ \langle buffer} = \langle < > > \\ \text{ \langle senderJdx} = 7 \\ \text{ \langle buffer} = \langle (1, 2, 3) \\ \text{ \langle ackSeqNum} = 0 \\ \text{ \langle senderJdx} = 7 \\ \text{ \langle buffer} = \langle (1, 2, 3) \\ \text{ \langle senderJdx} = \langle \text{ \langle senderJdx} = 7 \\ \text{ \langle buffer} = \langle (1, 2, 3) \\ \text{ \langle senderJdx} = \langle \text{ \langle senderJdx} = 7 \\ \text{ \langle senderJdx} = \langle \text{ \langle senderJdx} \\ \text{ \langle senderJdx} = \langle \text{ \langle senderJdx} \\ \text{ \langle senderJdx} = \langle \text{ \langle senderJdx} \\ \text{ \langle senderJdx} = \langle \text{ \langle Ack -> \langle A_R'',} \\ \langle sender -> \langle A_S'' \\ \text{ \langle sender -> \
\(\)\ \senderState = "WAITING" \(\)\ \ackWireIn = <<>> \(\)\ \ackWireOut = <<>> \(\)\ \dataWireOut = <<>> \(\)\ \(\)\ \attrict{ataWireIn} = <<>> \(\)\ \(\)\ \attrict{ataWireIn} = <<>> \(\)\ \(\)\ \attrict{output} = <<1, 3>> \(\)\ \(\)\ \attrict{ataWireIdx} = 3 \(\)\ \(\)\ \attrict{buffer} = <<>> \(\)\ \(\)\ \senderIdx = 3 \(\)\ \\ \attrict{buffer} = <<>> \(\)\ \(\)\ \senderPc = [SendSynAck -> "A_S", \(\)\ \(\)\ \ACK -> "A", \(\)\ \\ \Sender -> "A", \(\)\ \\ \Sender -> "A_R", \(\)\ \\ \Sender -> "A_S"] \(\)\ \(\)\ \receiverState = "WAITING"	\(\lambda \text{ackWir} \\ \lambda \text{dataWir} \\ \lambda \text{dataWir} \\ \lambda \text{dataWir} \\ \lambda \text{outpu} \\ \lambda \text{ackS} \\ \lambda \text{sen} \\ \lambda \text{senderPc} = [S] \\ \text{AC} \\ \text{SynA} \\ \text{ReceiveBur} \end{array}	ate = "WAITING" reIn = <<>> eOut = <<>> reOut = <<>> in = 0 ireIn = <<>> it = <<1>> it = <<1>> it = <<1>> if = < 1 if = 1 if	\(\) \(\senderState = \) \(\) \(\ack\WireIn = <<>> \) \(\ack\WireOut = <<>> \) \(\ack\WireOut = <<>> \) \(\ack\WireIn = <<>> \) \(\ack\WireOut = <<>> \) \(\ack\WireIn = <<1>> \) \(\ack\WireIn = <<1>> \) \(\ack\WireIn = <<>> \) \(\ack\WireIn = \) \(\ack\Wire = \) \(\(\) \(\senderState = \) \(\) \(\ack \)	\(\text{senderState} = \text{"WAITING"} \\ \text{ ackWireIn} = \langle < > > \\ \text{ ackWireOut} = \langle < > > \\ \text{ dataWireOut} = \langle < > > \\ \text{ dataWireIn} = \langle < > > \\ \text{ output} = \langle 1, 2 > > \\ \text{ ackSeqNum} = 0 \\ \text{ senderIdx} = 4 \\ \text{ buffer} = \langle < > > \\ \text{ senderPc} = [SendSynAck -> \text{"A_S",} \\ \text{ ACK } -> \text{"A_",} \\ \text{ SynAck } -> \text{"A_R",} \\ \text{ Sender } -> \text{"A_S"}] \\ \text{ receiverState} = \text{"WAITING"}	\(\) \(\sender		\(\lambda \text{ackWire} \) \(\lambda \text{ackWire} \) \(\lambda \text{dataWire} \) \(\lambda \text{dataWire} \) \(\lambda \text{dataWire} \) \(\lambda \text{output} \) \(\lambda \text{ackSe} \) \(\lambda \text{send} \) \(\lambda \text{send} \) \(\lambda \text{senderPc} = [See \) \(\lambda \text{SenderPc} = [See \) \(\lambda \text{SynAc} \) \(\text{SynAc} \) \(\text{ReceiveBur} \)	Tate = "WAITING" TireIn = <<>> reOut = <<>> ireOut = <<>> ireOut = <<>> A n = 0 VireIn = <<>> seqNum = 0 MireIdx = 8 Mer = <<>> SendSynAck -> "A_S", CK -> "A", Ack -> "A", MutFirst -> "A_R", MutFirst -> "A_S"] State = "WAITING"		\(\)\(\senderState = \)\(\)\(\ack\WireIn = <<>> \)\(\)\(\ack\WireOut = <<>> \)\(\)\(\ack\WireOut = <<>> \)\(\)\(\ack\WireIn = <<>> \)\(\)\(\ack\WireIn = <<>> \)\(\)\(\ack\WireIn = <<>> \)\(\)\(\ack\SeqNum = 0 \)			
/\ senderState = "WAITING"	\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireIn} = <<>> \\ \lambda \text{ackWireOut} = <<>> \\ \lambda \text{ataWireOut} = <<>> \\ \lambda \text{ataWireIn} = <<>> \\ \lambda \text{output} = <<1>> \\ \lambda \text{ackSeqNum} = 0 \\ \lambda \text{senderIdx} = 4 \\ \lambda \text{buffer} = <<>> \\ \lambda \text{senderPc} = [\text{SendSynAck} -> \text{"A_S"}, \\ \text{ACK} -> \text{"A_"}, \\ \text{Sender} \text{PceiveButFirst} -> \text{"A_R"}, \\ \text{Sender} -> \text{"A_S"}] \\ \lambda \text{receiverState} = \text{"WAITING"}	\(\)\ \senderState = "WAITING" \(\)\ \ackWireIn = <<>> \\ \)\ \(\)\ \ackWireOut = <<>> \\ \)\ \(\)\ \dataWireOut = <<>> \\ \)\ \(\)\ \dataWireIn = <<>> \\ \)\ \(\)\ \ackSeqNum = 0 \\ \(\)\ \ackSeqNum = 0 \\ \(\)\ \senderIdx = 4 \\ \(\)\ \buffer = <<>> \\ \\ \\ \text{AcK -> "A",} \\ \text{SenderPc} = [\text{ SendSynAck -> "A_S",} \\ \(\)\ \text{ACK -> "A",} \\ \\ \)\ \\ \text{SynAck -> "A, ",} \\ \\ \\ \text{ReceiveButFirst -> "A, R",} \\ \\ \\ \\ \text{Sender -> "A, S"] \\ \(\)\ \\ \text{receiverState} = "WAITING"	\(\) \(\text{senderState} = \) \"WAITING" \(\) \(\text{ackWireIn} = <<>> \) \(\) \(\text{dataWireOut} = <<>> \) \(\text{dataWireOut} = <<>> \) \(\text{dataWireIn} = <<>> \) \(\text{dataWireIn} = <<>> \) \(\text{dataWireIdx} = 4 \) \(\text{dataFeq} \) \(\text{senderIdx} = 4 \) \(\text{dataFeq} \) \(\text{senderPc} = [\text{SendSynAck} -> \] \(\text{ACK} -> \] \(\text{A} \), \(\text{SenderPc} = [\text{SendSynAck} -> \] \(\text{ACK} -> \] \(\text{A} \), \(\text{Sender} -> \] \(\text{A} \) \(\text{First} -> \) \(\text{A} \) \(\text{First} -> \text{A} \) \(F	\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireIn} = <<>> \\ \lambda \text{ackWireOut} = <<>> \\ \lambda \text{ataWireOut} = <<>> \\ \lambda \text{ataWireIn} = <<>> \\ \lambda \text{output} = <<1>> \\ \lambda \text{ackSeqNum} = 0 \\ \lambda \text{senderIdx} = 5 \\ \lambda \text{buffer} = <<>> \\ \lambda \text{senderPc} = [\text{SendSynAck} -> \text{"A_S",} \\ \text{ACK} -> \text{"A_",} \\ \text{SynAck} -> \text{"A_T",} \\ \text{ReceiveButFirst} -> \text{"A_R",} \\ \text{Sender} -> \text{"A_S"]} \\ \lambda \text{receiverState} = \text{"WAITING"}	\(\text{senderState} = \text{"WAITING"} \\ \text{ ackWireIn} = \leq \leq \rightarrow \\ \text{ ackWireOut} = \leq \leq \rightarrow \\ \text{ ataWireOut} = \leq \leq \rightarrow \\ \text{ ataWireIn} = \leq \leq \rightarrow \\ \text{ ackSeqNum} = 0 \\ \text{ ackSeqNum} = 0 \\ \text{ senderIdx} = 5 \\ \text{ buffer} = \leq \leq \rightarrow \\ \text{ senderPc} = [\text{ SendSynAck} - \rightarrow \text{ "A_S",} \\ \text{ Sender} - \text{ "A_S",} \\ \text{ Sender} - \text{ "A_S"}] \\ \text{ receiverState} = \text{ "WAITING"}	\(\) \(\senderState = \) \(\widehind{\text{WireIn}} = \leq \rightarrow \) \(\ack \widehindetwireOut = \leq \rightarrow \rightarrow \) \(\ack \widehindetwireOut = \leq \rightarrow \rightarrow \) \(\ack \widehindetwireIn = \leq \rightarrow \rightarrow \) \(\ack \widehindetwire \widehindetwire \widehindetwire \widehindetwire \rightarrow \rightarrow \widehindetwire	\(\) \(\senderState = \) \(\watersquare \) \(\senderState = \) \(\watersquare \) \(\senderState \) \(\(\text{senderState} = \text{"WAITING"} \\ \text{ ackWireIn} = \leqstarrow \rightarrow \text{ ackWireOut} = \leqstarrow \rightarrow \text{ dataWireOut} = \leqstarrow \rightarrow \text{ n} = 0 \\ \text{ dataWireIn} = \leqstarrow \rightarrow \text{ ackSeqNum} = 0 \\ \text{ ackSeqNum} = 0 \\ \text{ senderIdx} = 7 \\ \text{ buffer} = \leqstarrow \rightarrow \text{ Ack -> "A_S",} \\ \text{ ACK -> "A",} \\ \text{ SynAck -> "A_",} \\ \text{ ReceiveButFirst -> "A_R",} \\ \text{ Sender -> "A_S"]} \\ \text{ receiverState} = \text{ "WAITING"}	\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireIn} = \lambda \lambda \rangle \lambda \rangle \lambda \rangle \lambda \rangle \lambda \rangle \lambda \rangle \rangle \rangle \rangle \rangle \rangle \lambda \rangle \text{dataWireIn} = \lambda \lambda \rangle \ra		\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireIn} = <<>> \\ \lambda \text{ackWireOut} = <<>> \\ \lambda \text{ataWireOut} = <<>> \\ \lambda \text{ataWireIn} = <<>> \\ \lambda \text{output} = <<1, 2, 3>> \\ \lambda \text{ackSeqNum} = 0 \\ \lambda \text{senderIdx} = 8 \\ \lambda \text{buffer} = <<>> \\ \lambda \text{senderPc} = [SendSynAck -> \text{"A_S",} \\ \text{ACK } -> \text{"A_",} \\ \text{SynAck} -> \text{"A_",} \\ \text{Sender} -> \text{"A_s"} \] \\ \text{ReceiveButFirst} -> \text{"A_s"} \\ \text{Sender} -> \text{"A_s"} \] \\ \lambda \text{receiverState} = \text{"WAITING"}			
/\ senderState = "WAITING" /\ ackWireIn = <<>> /\ ackWireOut = <<>> /\ dataWireOut = <<>> /\ n = 7 /\ dataWireIn = <<>> /\ output = <<1, 3>> /\ ackSeqNum = 0 /\ senderIdx = 6 /\ buffer = <<>> /\ senderPc = [SendSynAck -> "A_S", ACK -> "A", SynAck -> "A", ReceiveButFirst -> "A_R", Sender -> "A s"] /\ receiverState = "WAITING"	\(\)\(\text{senderState} = \)\(\)\(\text{ackWireIn} = <<>> \)\(\)\(\text{ackWireOut} = <<>> \)\(\)\(\text{dataWireOut} = <<>> \)\(\text{dataWireIn} = <<>> \)\(\text{dataWireIn} = <<>> \)\(\text{dataWireIn} = 0 \)\(\text{dataSeqNum} = \)\(\text{dataSeqNum} = 0 \)\(\text{dataSeqNum} = 0 \)\(\text{dataSeqNum} = 0 \)\(\text{dataSeqNum} = 0 \)\(\text{dataSeqNum} = \)\(\text{dataSeqNum} = \)\(\text{dataSeqNum} = 0 \)\(\text{dataSeqNum} = \)\(\text{dataSeqNum} = 0 \)\(\text{dataSeqNum} = \)\(dataS	\(\text{senderState} = \text{"WAITING"} \\ \text{ ackWireIn} = \leqstarrow \rightarrow \\ \text{ ackWireOut} = \leqstarrow \rightarrow \\ \text{ dataWireOut} = \leqstarrow \rightarrow \\ \text{ n = 9} \\ \text{ dataWireIn} = \leqstarrow \rightarrow \\ \text{ ackSeqNum} = 0 \\ \text{ \text{ ackSeqNum} = 0} \\ \text{ \text{ senderIdx} = 6} \\ \text{ \text{ buffer} = \leqstarrow \rightarrow \rightarrow \text{ dcK -> "A",} \\ \text{ senderPc} = \text{ [SendSynAck -> "A_",} \\ \text{ SynAck -> "A_",} \\ \text{ Sender -> "A_s"]} \\ \text{ \text{ receiverState} = "WAITING"} \end{arrow}	\(\lambda \text{ackWireIn} = \) \(\lambda \text{ackWireOut} = \) \(\lambda \text{dataWireOut} = \) \(\lambda \text{dataWireIn} = \) \(\lambda \text{output} = << \) \(\lambda \text{ackSeqNum} \) \(\lambda \text{senderIdx} \) \(\lambda \text{buffer} = << \)	/ ackWireIn =	"WAITING" = <<>>	<pre></pre>	/AITING" /AckWire /A	vut = <<>>	\(\ackWireIn = <<>> \\ \ackWireOut = <<>> \\ \dataWireOut = <<>> \\ \dataWireOut = <<>> \\ \lambda \text{ dataWireIn } = <<>> \\ \lambda \text{ dataWireIn } = <<>> \\ \lambda \text{ output } = <<1, 2>> \\ \lambda \text{ ackSeqNum } = 0 \\ \lambda \text{ senderIdx } = 7 \\ \lambda \text{ buffer } = <<>> \\ \lambda \text{ senderPc } = [\text{ SendSynAck } -> \"A \\ \text{ ACK } -> \"A \\ \text{ R".} \\ \text{ ReceiveButFirst } -> \"A \\ \text{ R".} \end{array}	\(\text{senderState} = \text{"WAIT} \\ \text{ackWireIn} = \limits < \text{\$\times \text{dataWireOut} = \limits < \text{\$\times \text{dataWireIn} = \text{\$\text{dataWireIn} = \$\text{data	\(\lambda \text{ckWireIn} = \lambda \) \(\lambda \text{ackWireOut} = \) \(\lambda \text{dataWireOut} = \) \(\lambda \text{dataWireIn} = \) \(\lambda \text{dataWireIn} = \) \(\lambda \text{output} = <<1, \) \(\lambda \text{ackSeqNum} \) \(\lambda \text{senderIdx} = \) \(\lambda \text{buffer} = <<\\ \lambda \text{buffer} = <<\\ \lambda \text{SenderPc} = [SendSyn \) \(\lambda \text{CK} -> "A \)	/AITING" <<>>> <<>>> <<>>> > >> All the state of th		
\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireIn} = \lambda \lambda \\ \(\lambda \text{ackWireOut} = \lambda \lambda \\ \(\lambda \text{dataWireOut} = \lambda \lambda \\ \(\lambda \text{dataWireIn} = \lambda \lambda \\ \(\lambda \text{output} = \lambda \lambda \lambda \text{seqNum} = 0 \\ \(\lambda \text{senderIdx} = 7 \\ \(\lambda \text{buffer} = \lambda \lambda \rangle \text{senderIdx} = 7 \\ \(\lambda \text{buffer} = \lambda \lambda \rangle \text{senderIdx} = 7 \\ \(\lambda \text{buffer} = \lambda \lambda \rangle \rangle \text{"A_S",} \\ \(\lambda \text{ceiveButFirst} \rangle \rangle \text{"A_R",} \\ \(\lambda \text{sender} \rangle \rangle \text{"A_R",} \\ \(\lambda \text{sender} \rangle \rangle \text{"A_R",} \\ \(\lambda \text{sender} \rangle \rangle \text{"A_ITING"} \)	\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireIn} = \lambda \lambda \\ \text{ackWireOut} = \lambda \lambda \\ \lambda \text{ataWireIn} = \lambda \lambda \\ \lambda \text{ataWireIn} = \lambda \lambda \\ \text{ackSeqNum} = 0 \\ \lambda \text{ ackSeqNum} = 0 \\ \lambda \text{ senderIdx} = 8 \\ \lambda \text{ buffer} = \lambda \lambda \\ \text{ buffer} = \lambda \lambda \\ \text{ senderPc} = [\text{ SendSynAck} -> \text{ "A_S",} \\ \text{ ACK} -> \text{ "A_",} \\ \text{ Sender} -> \text{ "A_",} \\ \text{ Sender} -> \text{ "A_S"} \\ \text{ receiveButFirst} -> \text{ "A_R",} \\ \text{ Sender} -> \text{ "A_S"} \\ \text{ \text{ receiverState}} = \text{ "WAITING"} \end{array}		\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireIn} = <<>> \\ \lambda \text{ackWireOut} = <<>> \\ \lambda \text{dataWireOut} = <<>> \\ \lambda \text{dataWireIn} = <<>> \\ \lambda \text{ackSeqNum} = 0 \\ \lambda \text{senderIdx} = 6 \\ \lambda \text{buffer} = <<>> \\ \lambda \text{senderIdx} = 6 \\ \lambda \text{buffer} = <<>> \\ \lambda \text{senderPc} = [\text{SendSynAck} -> \text{"A}, \\ \text{SynAck} -> \text{"A}, \\ \text{ReceiveButFirst} -> \text{"A}, \\ \text{Sender} ->	\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireIn} = <<>> \\ \lambda \text{ackWireOut} = <<>> \\ \lambda \text{dataWireOut} = <<>> \\ \lambda \text{dataWireIn} = <<>> \\ \lambda \text{output} = <<1>> \\ \lambda \text{ackSeqNum} = 0 \\ \lambda \text{senderIdx} = 6 \\ \lambda \text{buffer} = <<>> \\ \lambda \text{senderPc} = [\text{SendSynAck} -> \text{"A_S",} \\ \text{ACK} -> \text{"A_",} \\ \text{SynAck} -> \text{"A_R",} \\ \text{Sender} -> \text{"A_S"}] \\ \lambda \text{receiveButFirst} -> \text{"A_S"}] \\ \lambda \text{receiverState} = \text{"WAITING"}	\(\text{senderState} = \text{"WAITING"} \\ \text{ ackWireIn} = \leq \leq \rightarrow \\ \text{ ackWireOut} = \leq \leq \rightarrow \\ \text{ ataWireIn} = \leq \leq \rightarrow \\ \text{ ataWireIn} = \leq \leq \rightarrow \\ \text{ ackSeqNum} = 0 \\ \text{ ackSeqNum} = 0 \\ \text{ senderIdx} = 6 \\ \text{ buffer} = \leq \leq \rightarrow \text{ ACK} \rightarrow \text{ "A_",} \\ \text{ Sender -> "A_",} \\ \text{ ReceiveButFirst -> "A_R",} \\ \text{ Sender -> "A_s"] \\ \text{ receiverState} = \text{ "WAITING"}		\(\text{senderState} = \text{"WAITING"} \\ \text{ ackWireIn} = \leqstarrow \rightarrow \text{ dataWireOut} = \leqstarrow \rightarrow \text{ dataWireIn} = \leqstarrow \rightarrow \righta	\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireIn} = \lambda \lambda \text{ \(\lambda \text{ackWireOut} = \lambda \lambda \text{ \(\lambda \text{dataWireIn} = \lambda \lambda \text{ \(\lambda \text{dataWireIn} = \lambda \lambda \text{ \(\lambda \text{dataWireIn} = \lambda \lambda \text{ \(\lambda \text{catack} \text{vinus} = 0 \\ \lambda \text{ senderIdx} = 8 \\ \lambda \text{ buffer} = \lambda \lambda \text{ senderIdx} = 8 \\ \lambda \text{ buffer} = \lambda \lambda \text{ \(\lambda \text{sender} \text{ValTING} \rangle \text{ \(\lambda \text{sender} \text{ValTING} \rangle \text{ \(\lambda \text{sender} \rangle \rangle \rangle \rangle \text{A} \rangle \rangle \rangle \text{A} \rangle \rangle \rangle \text{A} \rangle \rangle \rangle \rangle \text{A} \rangle \rang	\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireIn} = <<>> \\ \lambda \text{ackWireOut} = <<>> \\ \lambda \text{dataWireOut} = <<>> \\ \lambda \text{dataWireIn} = <<>> \\ \lambda \text{output} = <<1, 2>> \\ \lambda \text{ackSeqNum} = 0 \\ \lambda \text{senderIdx} = 9 \\ \lambda \text{buffer} = <<>> \\ \lambda \text{senderIdx} = \text{SenderNack} \cdot -> \text{"A} \\ \text{ACK} \cdot -> \text{"A} \\ \text{ACK} \cdot -> \text{"A} \\ \text{ReceiveButFirst} \cdot -> \text{"A} \\ \text{ReceiveButFirst} \cdot -> \text{"A} \\ \text{R}'', \\ \text{Sender} \cdot -> \text{"A} \\ \tex					
\(\text{\section} \) \(\text{\section} \) \(\text{\ackWireIn} = <<>> \) \(\text{\ackWireOut} = <<>> \) \(\text{\ackWireOut} = <<>> \) \(\text{\ackWireOut} = <<>> \) \(\text{\ackWireIn} = <<>> \) \(\text{\ackWireIn} = <<>> \) \(\text{\ackSeqNum} = 0 \) \(\text{\ackSeqNack} \] \(\text{\ackSeqNack} \) \(\text{\ackSeqNack} \] \	/\ senderState = "WAITING"	\(\lambda \text{senderState} = \text{"WAITING"} \\ \lambda \text{ackWireIn} = \lambda \lambda \\ \(\lambda \text{ackWireOut} = \lambda \lambda \\ \(\lambda \text{nata} = \lambda \lambda \\ \(\lambda \text{nata} \text{vicIn} = \lambda \lambda \\ \(\lambda \text{ata} \text{wireIn} = \lambda \lambda \\ \(\lambda \text{ata} \text{vicIn} = \lambda \lambda \\ \(\lambda \text{ata} \text{wireIn} = \lambda \lambda \\ \(\lambda \text{senderIdx} = \text{8} \\ \(\lambda \text{senderPc} = \text{SendSynAck} \cdot \rangle \ran				\(\text{senderState} = \text{"WAITING"} \\ \text{ \ackWireIn} = \leq \leq \rightarrow \\ \text{ \ackWireOut} = \leq \leq \rightarrow \\ \text{ \ackWireIn} = \leq \leq \rightarrow \\ \text{ \ackWireOut} = \leq \leq \rightarrow \\ \text{ \ackSeqNum} = 0 \\ \text{ \ackSeqNum} = 0 \\ \text{ \senderIdx} = 9 \\ \text{ \begin{array}{c} \begin{array}{c} \text{ \senderIdx} = 9 \\ \text{ \senderPc} = [\text{ \sendsynAck} -> \text{ \sender} \rightarrow \\ \text{ \senderPc} = [\text{ \sendsynAck} -> \text{ \sender} \\ \s								
		\(\lambda \text{ackWireIn} = \lambda \lambda \rangle \text{ackWireOut} = \lambda \lambda \rangle \text{dataWireOut} = \lambda \lambda \rangle \text{dataWireIn} = \lambda \lambda \rangle \text{dataWireIn} = \lambda \lambda \rangle \text{senderIdx} = 9 \\ \lambda \text{senderIdx} = 9 \\ \lambda \text{buffer} = \lambda \lambda \rangle \text{senderIdx} = 9 \\ \lambda \text{senderJack} \rangle \lambda \rangle \text{senderJack} \rangle \rangle \lambda \rangle \text{senderJack} \rangle \												