The 'Rendezvous' code

Problem: Two cogs have been running each their own code for some time. A critical piece of code requires that the two cogs run exactly in step - from a certain point onwards we know that instruction 1 in cog A will execute simultaneously with instruction 1 in cog B.

I first of all tried this code:-

Cog A	**Cog B**
Code ...	
...Code	Code ...
COGATN CogB	...Code
NOP	**WAITATN**
Instruction 1	Instruction 1
Instruction 2	Instruction 2

As long as Cog B arrives early, the cogs synchronize as required. Cog B executes WAITATN and waits, Cog A arrives and executes COGATN to release Cog B and they set off together to execute Instruction 1 on exactly the same clock edge. However, if Cog A arrives early, synchronism is not achieved. Cog A executes COGATN and carries on. Later Cog B executes WAITATN, which cancels the ATN flag and Cog B just carries on – no waiting.

Mm - this seems to be half the solution, so what if we added the same code to work in the opposite direction?

Cog A	**Cog B**
Code ...	Code ...
...Code	**WAITATN**
COGATN CogB	**COGATN CogA**
WAITATN	**NOP**
Instruction 1	Instruction 1
Instruction 2	Instruction 2

Bingo! No matter which cog arrives first, they will wait for the other and then set off together in synchronism. Obviously, synchronism will be lost if interrupts are enabled, also when instructions with variable timing are executed e.g. rdlong, wrlong

So this is my ‘Rendezvous’ code and it’s copyright free to anyone who wishes to use it. It comes with no guarantee that it will always work, and it’s up to the programmer to determine that it’s fit for his purpose.

Bob Edwards, Ham radio call G4BBY, SW England, January 2026