AUTOMOTIVE WAKEUP METHODS

Natalie Wienckowski

September 2017 IEEE802.3 Interim

GENERAL MOTORS

AGENDA

GENERAL WAKEUP REQUIREMENTS

Low Current Draw

- Vehicle “sleeping”, no Ethernet traffic: 30 to 50 µA per PHY from Battery
- Vehicle “awake”, some Ethernet traffic but specific ECU not required: 750 µA max, total from all sources

Fast wakeup distribution

- Recognized by a sleeping node within 2 ms, 1 ms for an awake node
- Passed on by a sleeping switch to all other switch ports within 15 ms, 2 ms for an awake switch
- Note: Shall not require link to receive or send wakeup

ECU Initialization and link-up occur in parallel after wakeup is received

OPTIONS FOR WAKEUP

Wake on Traffic: All traffic on the bus is a wakeup. As long as there is any communication, all ECUs will be awake. Similar to CAN WUP or LIN Wakeup.

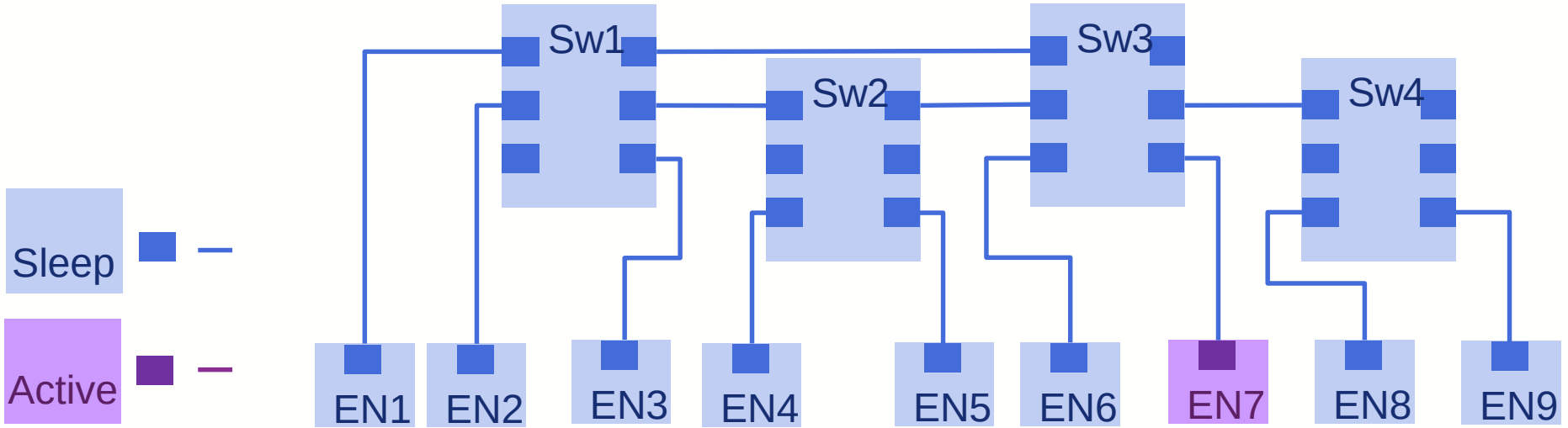
PN Wakeup: Data is embedded in the wakeup message which each device uses to determine if they need to wake up or not. This does not require any micro interaction and is done at the PHY level. See PN CAN Example.

Pseudo PN Wakeup: A single wakeup signal is used. This may be a specific pattern or voltage. All ECUs are woken up by this signal. Messages are sent which the micros can use to determine if the ECU needs to stay awake or can go back to sleep. See swCAN Example.

WAKE ON TRAFFIC

EN7 wakes up and starts sending “messages”

All Switches forward all “messages”

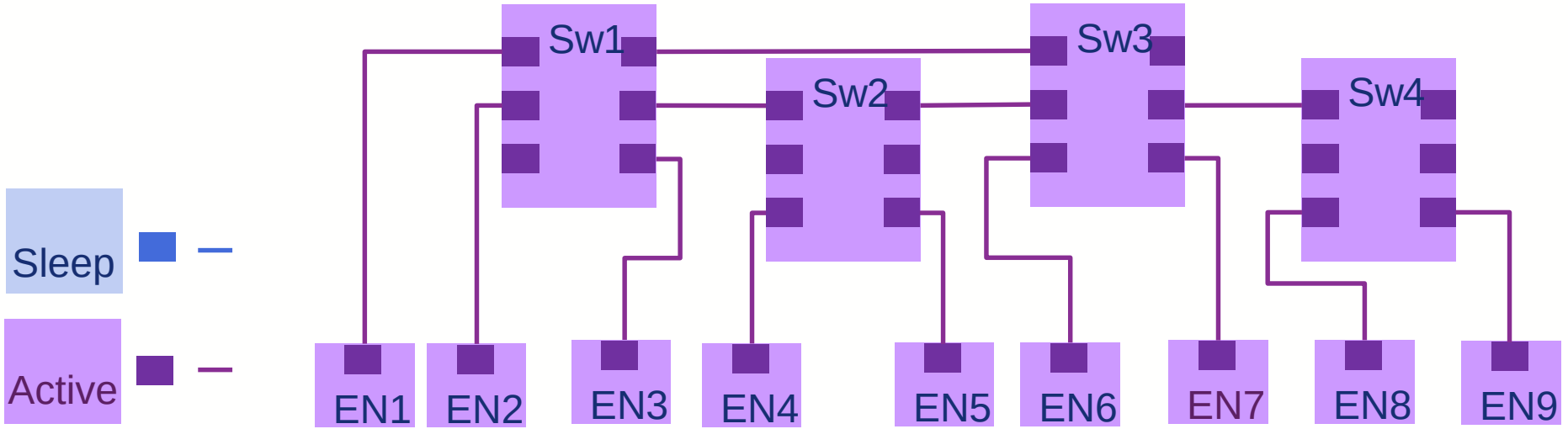


WAKE ON TRAFFIC

All End Nodes wake up

Micros are started and PHYs/Switches start link-up

All ECUs remain awake as long as there is any traffic on the network

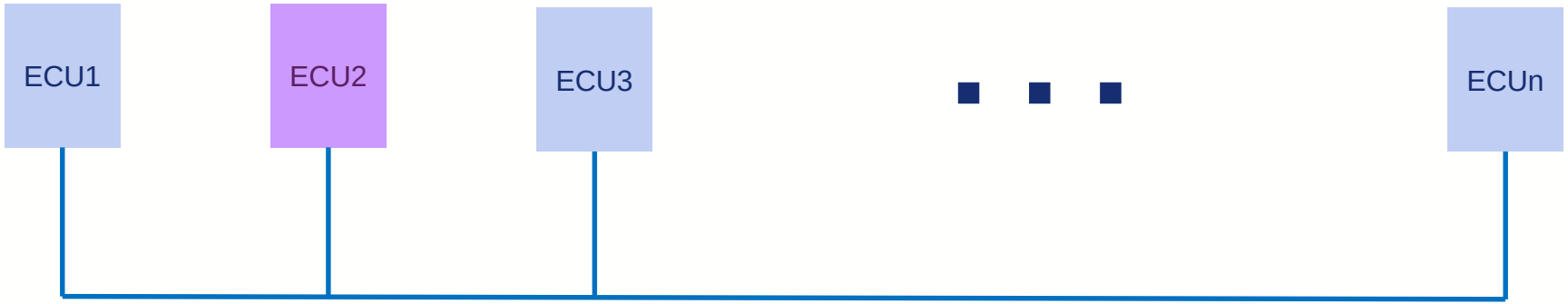


WAKE ON TRAFFIC EXAMPLE (CAN OR LIN)

ECU2 Wakes up

ECU2 transmits on the bus, may be any message or a specific pattern

All ECUs are electrically connected so they see the signal at the same time, neglecting delay in the wire (~ 6 ns/m)



WAKE ON TRAFFIC EXAMPLE (CAN OR LIN)

Transceivers (PHYs) receive bus signal

Indicate Wake up to ECU assuming valid wake up signal (usually single or multiple pulses of specified duration within a specified time)

Valid Wake up signal (WUP for CAN) causes all ECUs to Wake and prepare for communication session

See ISO11898-2:2016 for WUP details

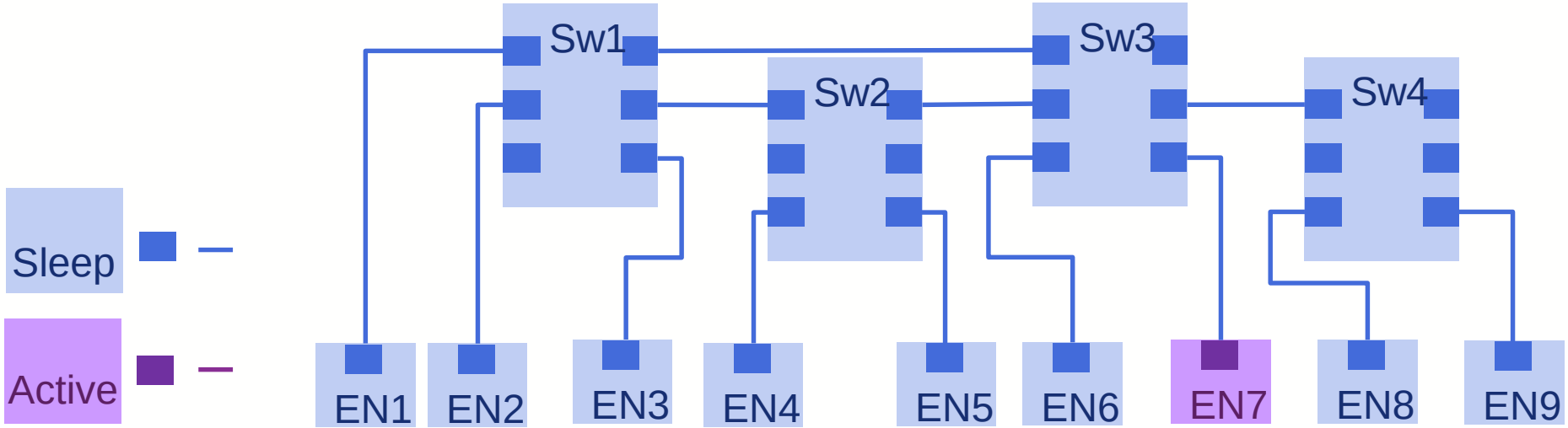
PARTIAL NETWORK WAKEUP

All Switches forward all “wakeup” messages

End Nodes are configured to wakeup only on specific message(s)/pattern(s)

Switches only turn on the ports needed for communication.

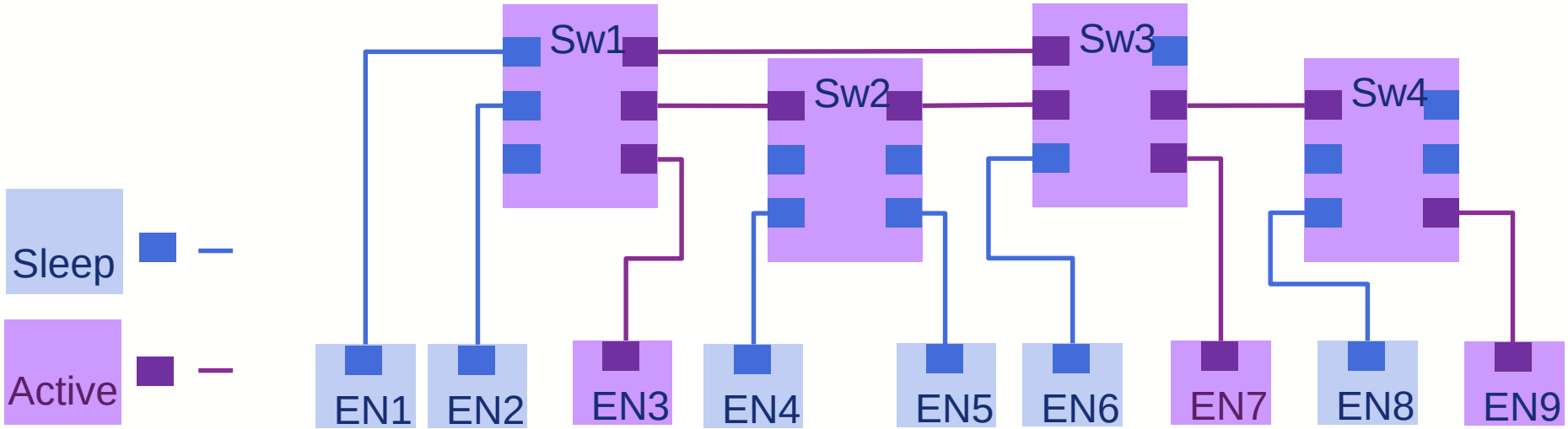
EN7 wakes up, it starts a virtual network which includes EN9 and EN3



PARTIAL NETWORK WAKEUP

EN7 wakes up, it starts a virtual network which includes EN9 and EN3

This is the result of the wake up sequence



PN CAN EXAMPLE

2 Stage Wakeup

1st Stage

- WUP (Wake Up Pattern)
  - In most, if not all, messages
  - Includes filters to ensure noise is not detected as a wakeup
  - Causes bus biasing to be turned on in the transceiver
  - Starts clock for detection of WUF used in 2nd Stage
- 2nd Stage
  - WUF (Wake Up Frame)
  - Specific ID(s) and data bit(s) that Wakeup individual ECUs or sets of ECUs.

See ISO11898-2:2016 for more details

PN CAN EXAMPLE – 2ND STAGE

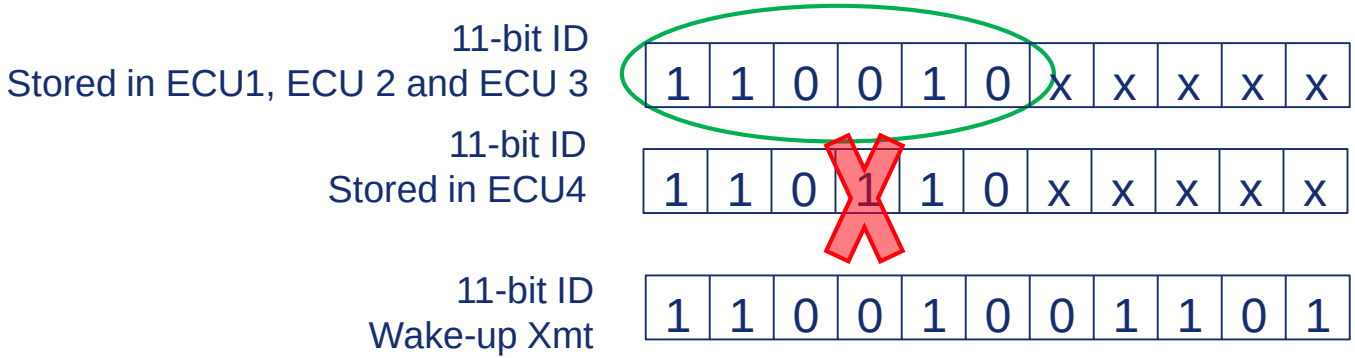
Use an 11-bit ID for the wake-up message.

The first 6 bits are the base ID used for all wake-up messages.

The next 5 bits are used to distinguish between the different wake-up sources to meet the CAN protocol requirement that multiple ECUs can't transmit the same ID.

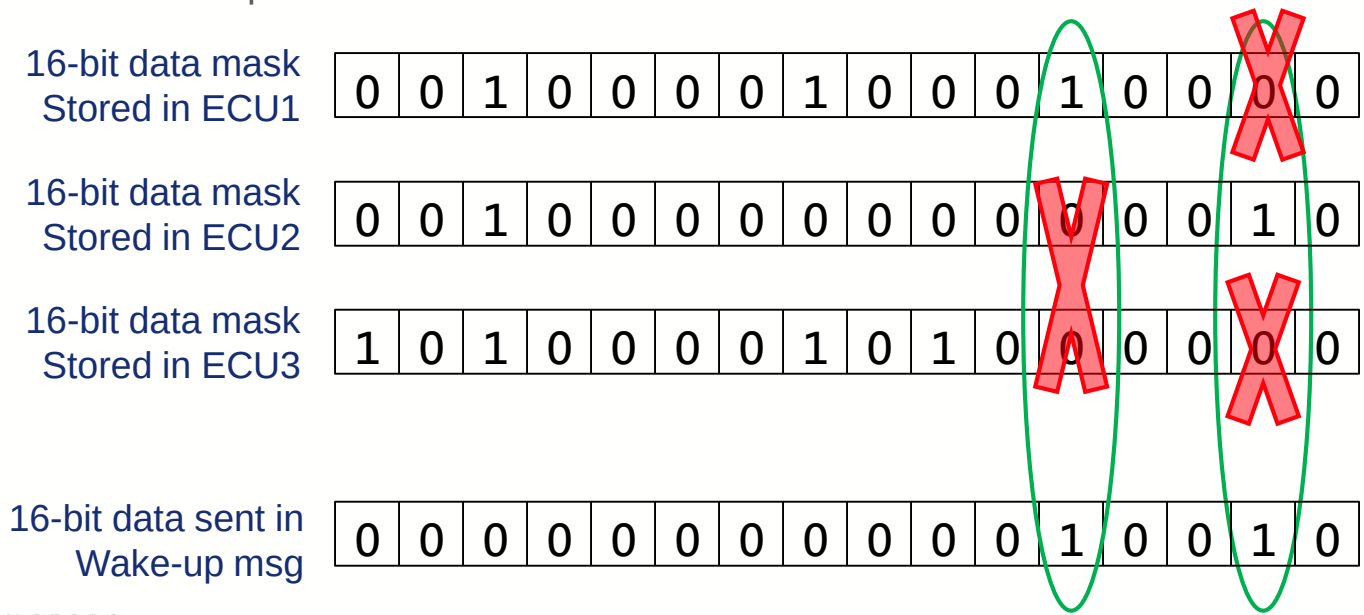
ECU's 1, 2 and 3 may wake up (depends on data) as the first 6 bits of the sent ID match all of the bits that aren't "x" in the stored ID.

ECU4 will not wake up as bit 8 of the sent ID doesn't match the bits stored ID bit.



PN CAN EXAMPLE – 2ND STAGE, CONTINUED

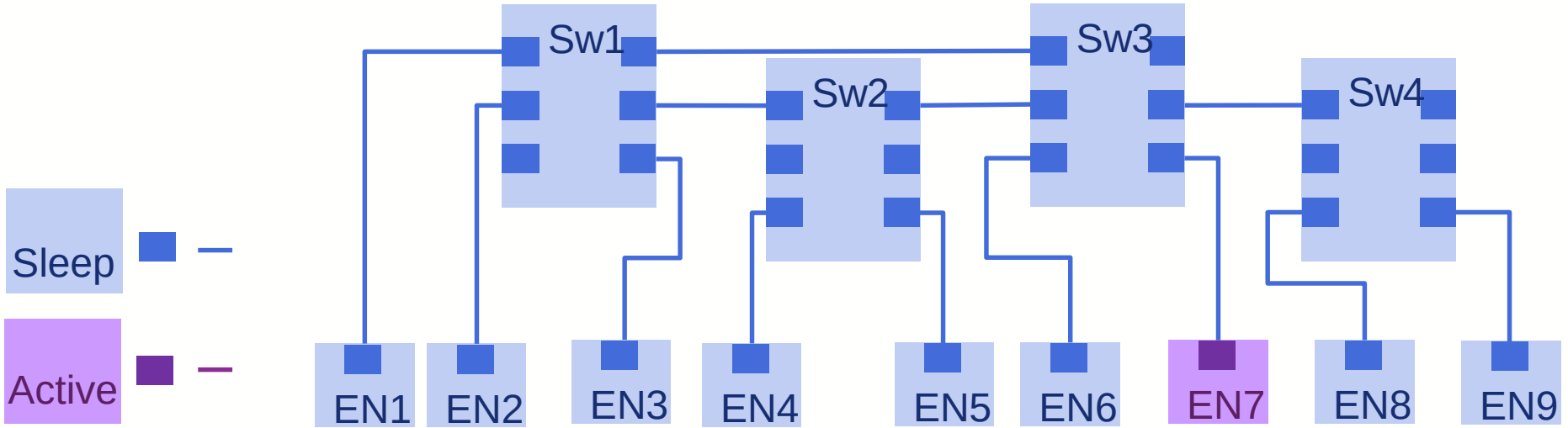
Use 16-bit data for the active VNs for this example.
Example assumes ID matched as shown in earlier slides.
ECU 1 wakes up because bit 4 matches.
ECU2 wakes up because bit 1 matches.
ECU3 doesn't wake up.



PSEUDO PARTIAL NETWORK WAKEUP

EN7 wakes up

All Switches forward all “wakeup” messages

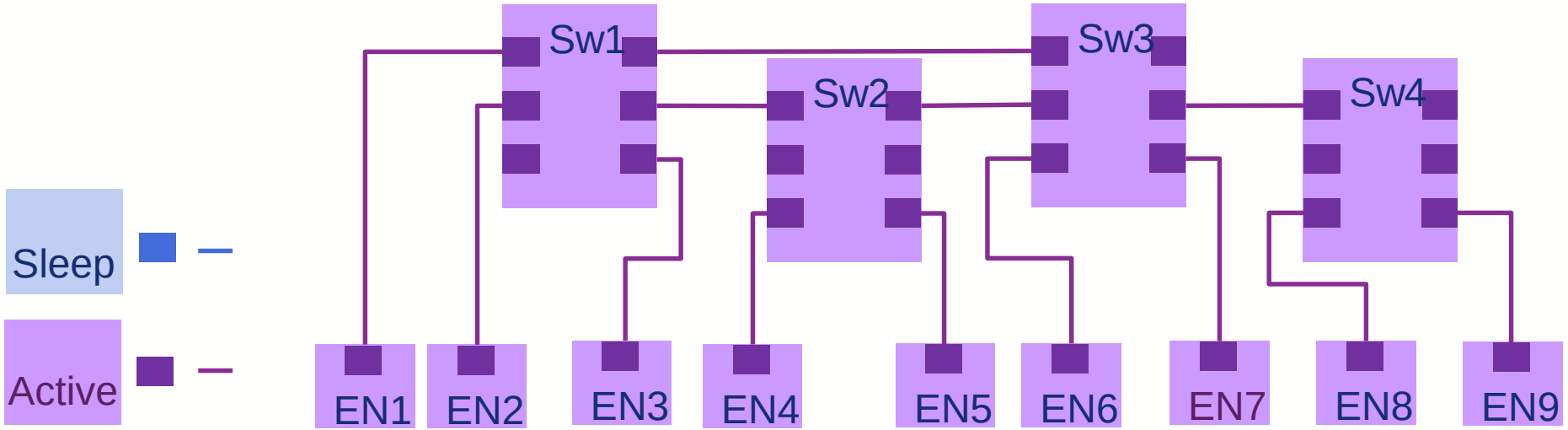


PSEUDO PARTIAL NETWORK WAKEUP

All End Nodes wake up

Micros are started and PHYs/Switches start link-up

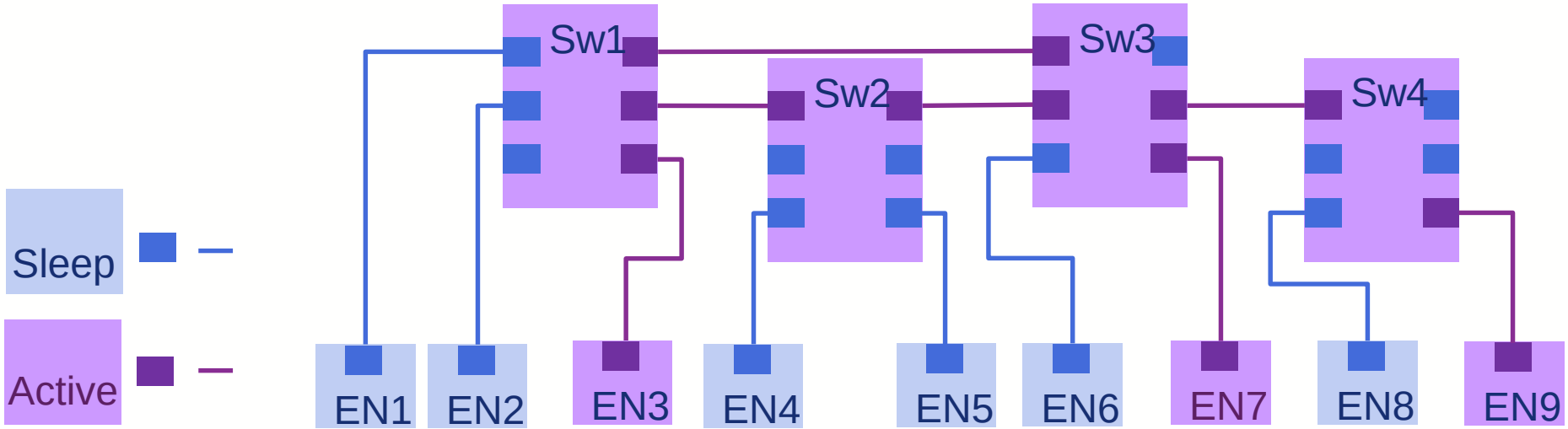
Virtual Network Management messages are sent which determine which ECUs are required for the desired operation.



PSEUDO PARTIAL NETWORK WAKEUP

EN7 wakes up, it starts a virtual network which includes EN9 and EN3

This is the result of the wake up sequence



SWCAN EXAMPLE

Purpose

- Supports partial networks: allows some nodes to sleep while others communicate

Implementation

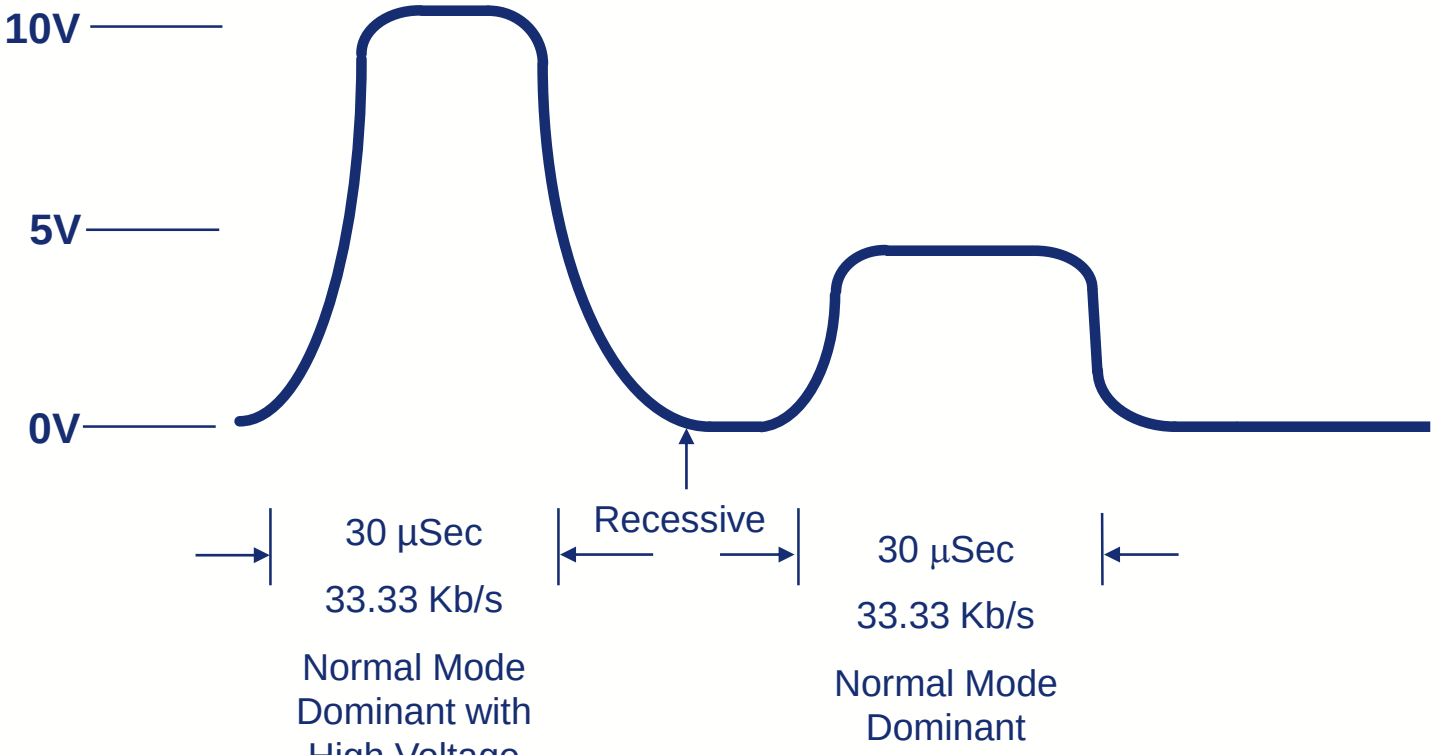
- Speed: 33.33Kbps
- Voltage level: ~10 Volts
- Edges controlled to meet EMC requirements

Activation

- Nodes are awakened via HVWU
- Nodes determine if participating (VNMF)
- Non-participating ECUs go to sleep
- Participating ECUs communicate at normal levels

See GMW3089 for more details

SWCAN EXAMPLE



SUMMARY

Wakeup type	Typ. Sleep Current (µA) *No Traffic on bus*	Typ. Wake Detect Current (µA) *Traffic on bus*	Summary		
			i	HW	SW
Wake on Traffic	30	NA	-	+	+
Partial Network	30	750	+	-	+
Pseudo Partial Network	30	Varies*	o	o	-

* Wakeup signal brings up entire ECU to understand messages, total average current depends on time awake and time between wakeup signals.

KEY: + low impact, o moderate impact, - high impact

QUESTIONS?