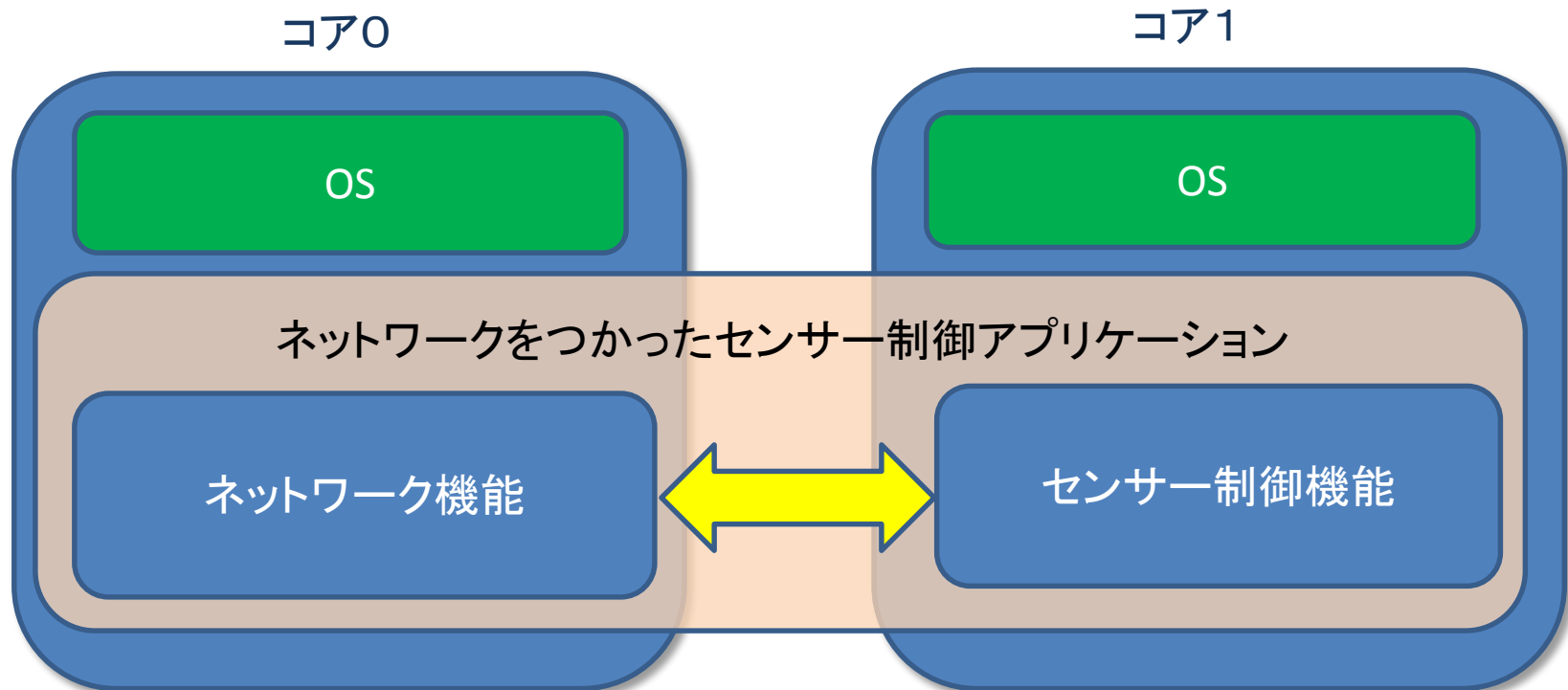


μC3/Standard+M簡易資料



AMP (Asymmetric Multiple Processor)とは？

コアごとに別の機能（処理）を専任する⇔(SMP:同じ処理を適時分担)

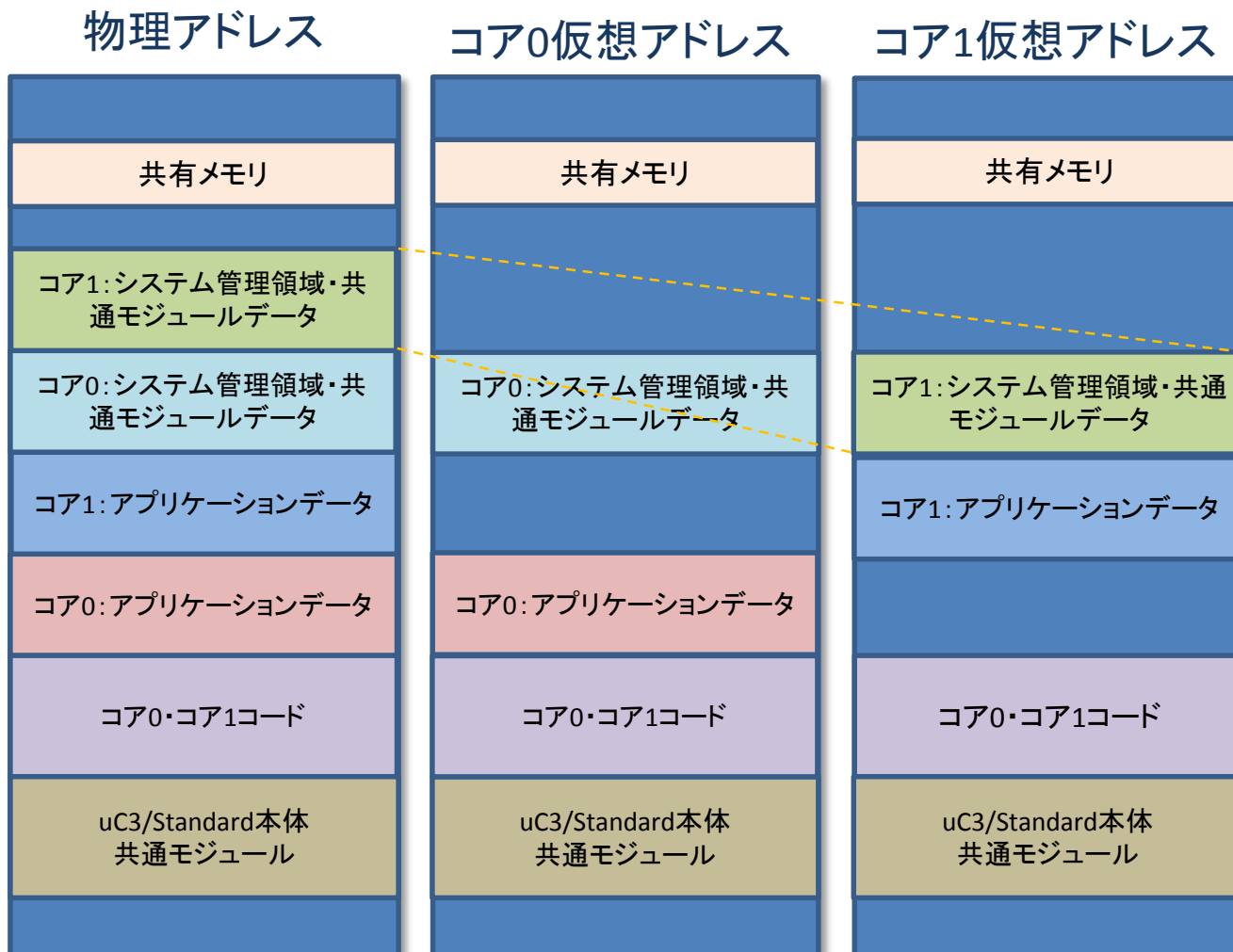


(利点)

- 処理の分担が発生するSMP形式と違い、高いリアルタイム性能が発揮できる。
- 各コアに機能が割り当てられるため、機能分担によって性能向上が狙える。

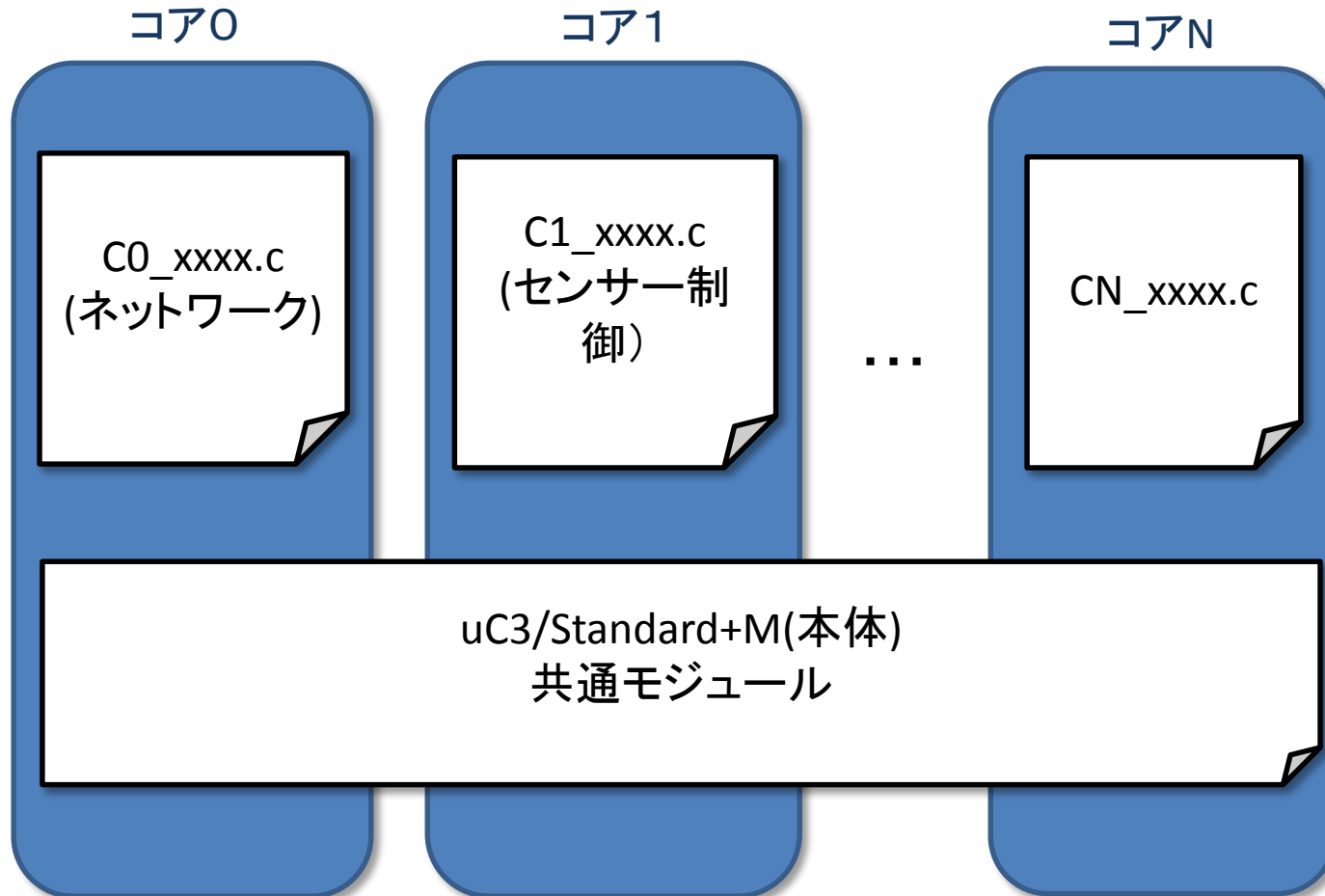
メモリマップの管理は？

コアごとにMMUテーブルを編集。MMUテーブルとリンクスクリプトによって管理する。



処理(ソースコード)の管理は？

コア毎の処理はCn_xxxx.c(n=0~4)などのC言語のソース内に記述
(それ以外は)共通モジュールとしてどのコアからも利用可能

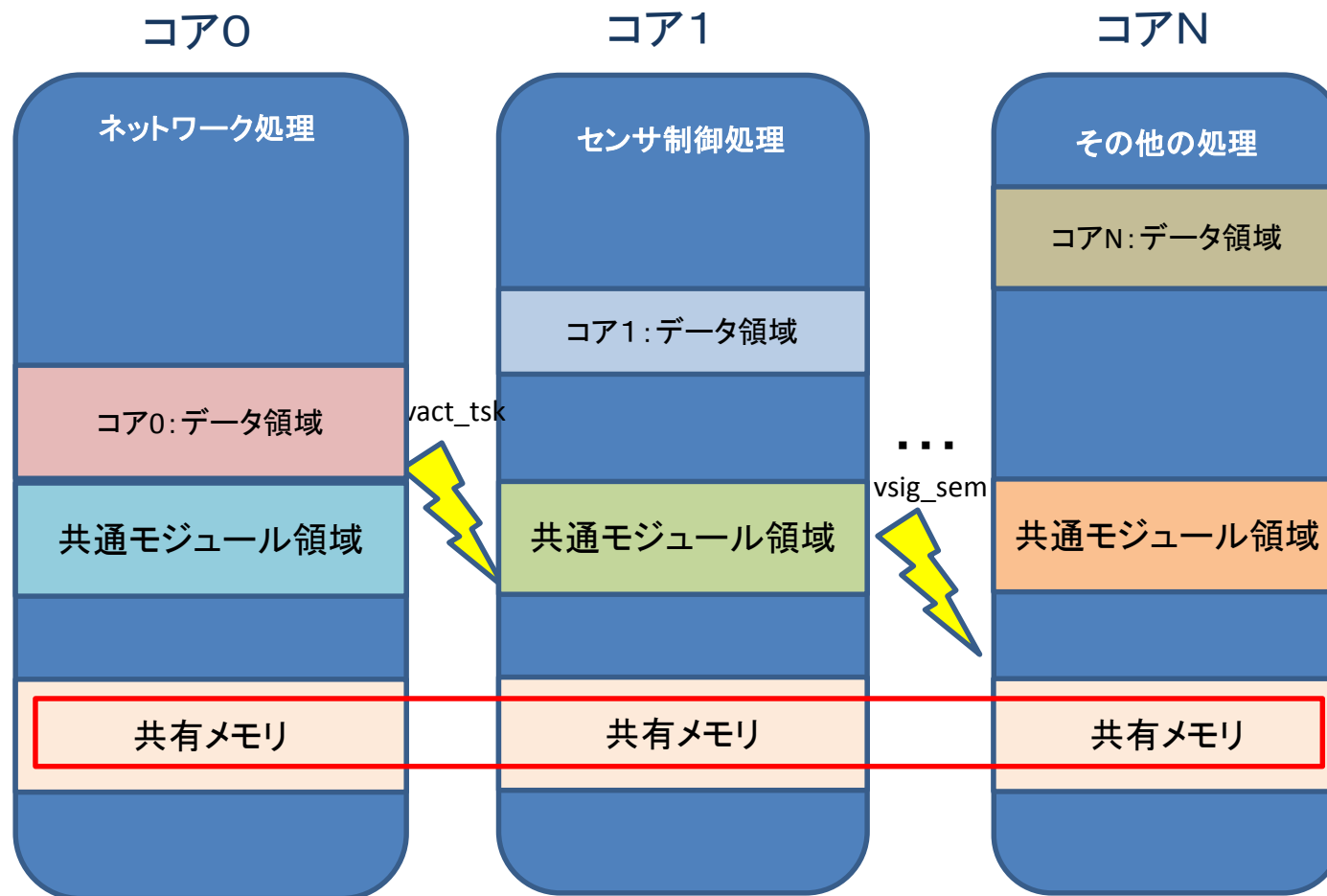


コア間の制御やデータのやりとりはどうするのか？

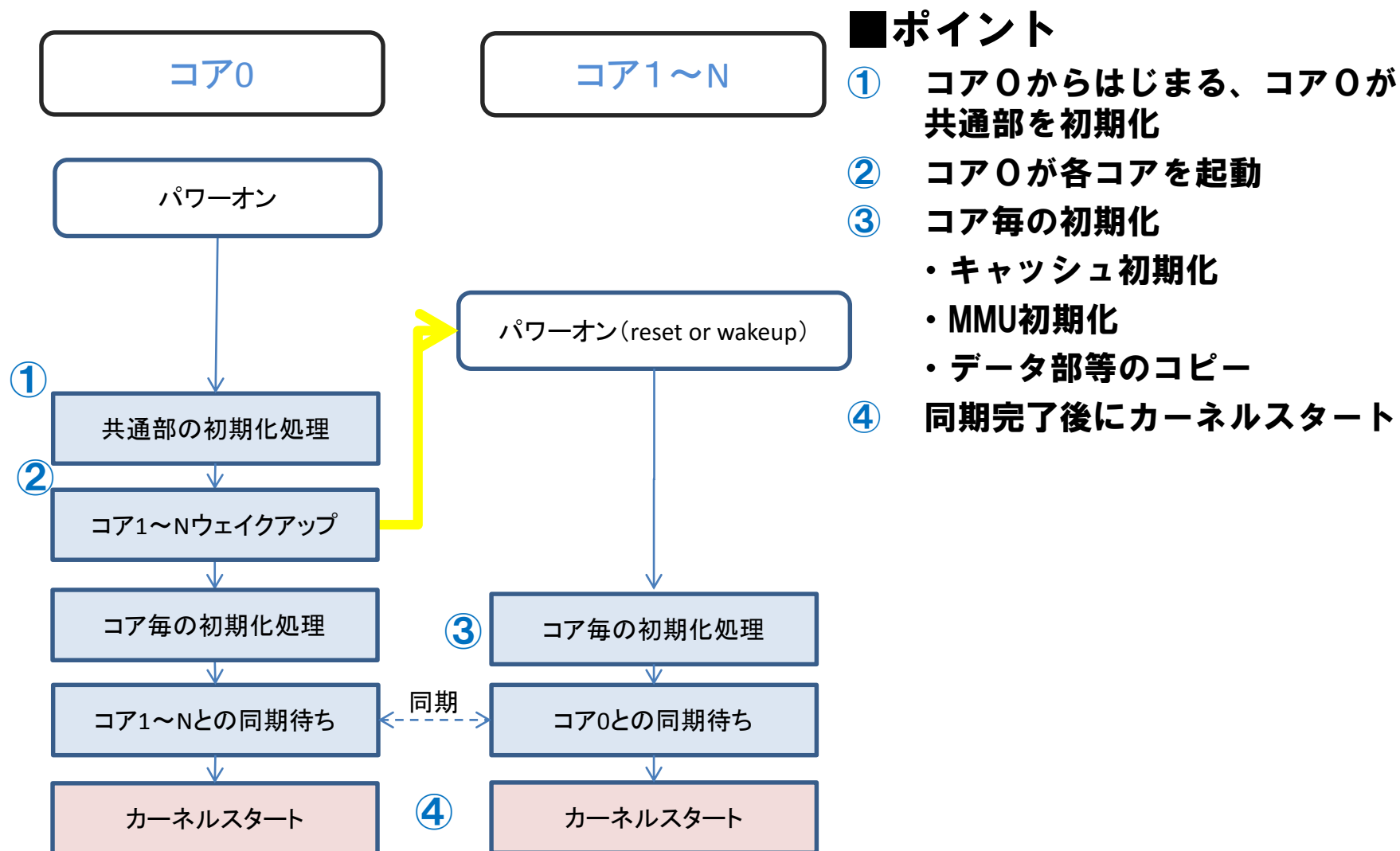
制御: 専用のシステムコールを使う

(vact_tsk, vset_flg, vsig_sem, vpsnd_dtq, ...)

データのやりとり: 共有メモリ領域を使う



スタートアップはどうなっているのか？



+M独自のシステムコール(コア間制御用)

機能	システムコール名	説明
タスク管理機能	vact_tsk	タスクの起動
	ivact_tsk	
	vsta_tsk	タスクの起動(起動コード指定)
タスク付属同期機能	vwup_tsk	タスクの起床
	ivwup_tsk	
	vrel_wai	待ち状態の強制解除
	ivrel_wai	
セマフォ	vsig_sem	セマフォ資源の返却
	ivsig_sem	
	vpol_sem	セマフォ資源の獲得(ポーリング)
イベントフラグ	vset_flg	イベントフラグのセット
	ivset_flg	
	vclr_flg	イベントフラグのクリア
	vpol_flg	イベントフラグ待ち(ポーリング)
データキュー	vpsnd_dtq	データキューへの送信(ポーリング)
	ivpsnd_dtq	
	vfsnd_dtq	データキューへの強制送信
	ivfsnd_dtq	
	vprcv_dtq	データキューからの受信(ポーリング)
時間管理機能	ivsig_tim	タイムチェックの供給
システム状態管理機能	vrot_rdq	タスクの優先順位の回転
	ivrot_rdq	
	get_cid	コアIDの取得

- **get_cid以外はすべてコアIDが指定でき、他コアに対して発行可能**
 - 両コア共有のセクションに管理情報を置いてから、IPL(コア間割込み)を発行して他コアに通知