ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Міністерство освіти та науки України
Харківський національний університет радіоелектроніки

кафедра ЕОМ

КВАЛІФІКАЦІЙНА РОБОТА

Методи створення архітектурних специфікацій обчислювальних комплексів на базі систем на кристалі

виконав: ст. гр. КСМзм-21-1 Моісєєв Д.С.
керівник: зав.каф. Коваленко А.А.

Мета та завдання

Метою роботи є аналіз методів створення архітектурних специфікацій обчислювальних комплексів на базі SoC.

Об'єктом дослідження є методи забезпечення функціонування обчислювальних комплексів на SoC.

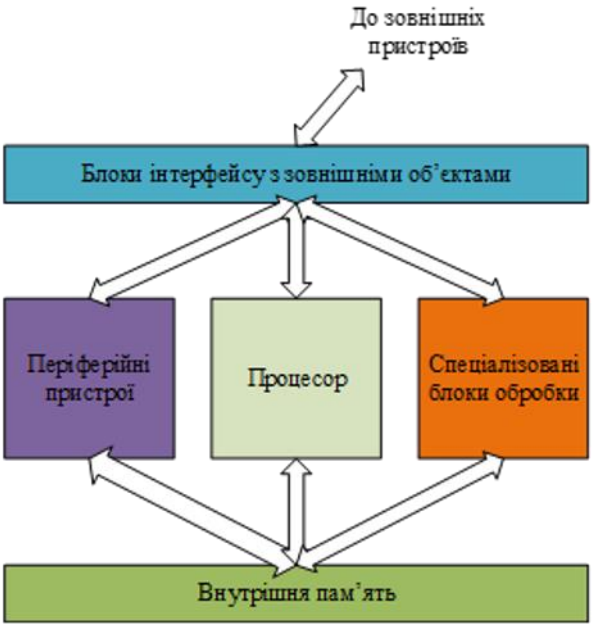
Завдання:

-проведення аналізу існуючих методів створення архітектурних специфікацій обчислювальних комплексів на базі SoC;

-проведення порівняння різних форматів архітектурних специфікацій SoC;

-розробка методу створення та використання архітектурних специфікацій обчислювальних комплексів на базі SoC.

Структурна схема SoC в загальному вигляді



3

Класифікація SoC в загальному вигляді

1. SoC побудовані навколо мікроконтролера
2. SoC побудовані навколо мікропроцесора
3. Спеціалізовані прикладні інтегральні схеми SoC, призначені для конкретних застосувань:

- Programmable SoCs (PSoC);
- Multiprocessor System-on-Chip, (MPSoC).

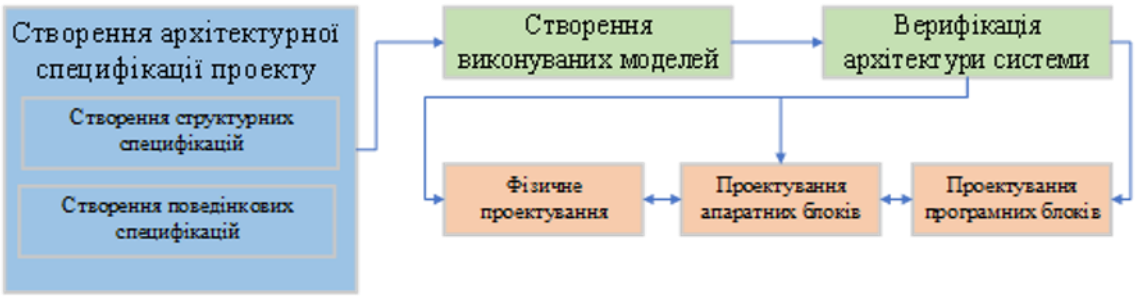
4

Способи вирішення вибору нових мов опису системного рівня та методології проектування

- ✓ Використання мов опису апаратури з деякими доповненнями, як, наприклад, розширення мови Verilog до SystemVerilog .
- ✓ Адаптація мов програмування високого рівня (C / C ++, Java) для опису архітектурних специфікацій систем на кристалі.
- ✓ Створення принципово нових мов системного рівня спеціально для проектування систем на кристалі.
- ✓ Розширення можливостей існуючих мов високорівневого моделювання та застосування їх разом з іншими мовами нижчого рівня для опису архітектурних специфікацій систем.

5

Процес розробки SoC



6

Проблеми сучасного процесу розробки SoC

Проблеми сучасного процесу розробки СнК

Апаратне і програмне забезпечення системи найчастіше розробляється різними командами з вузькою спеціалізацією, і інтеграція їх починається тільки на пізніх етапах проектування системи. Ця ситуація веде до неможливості спільного аналізу апаратних і програмних компонентів системи на ранніх етапах проектування.	При створенні архітектурних специфікацій системи необхідно строго стежити за інкапсуляцією компонентів системи і повним і строгим описом їх інтерфейсів.	Процес розробки виконуваних моделей архітектури складних систем на кристалі загального призначення пов'язан зі значними труднощами, і доводиться починати розробку компонентів системи до готовності архітектурної моделі.	Архітектурні специфікації в більшості своїй написані на природних мовах, і їм притаманні всі недоліки цих мов.

7

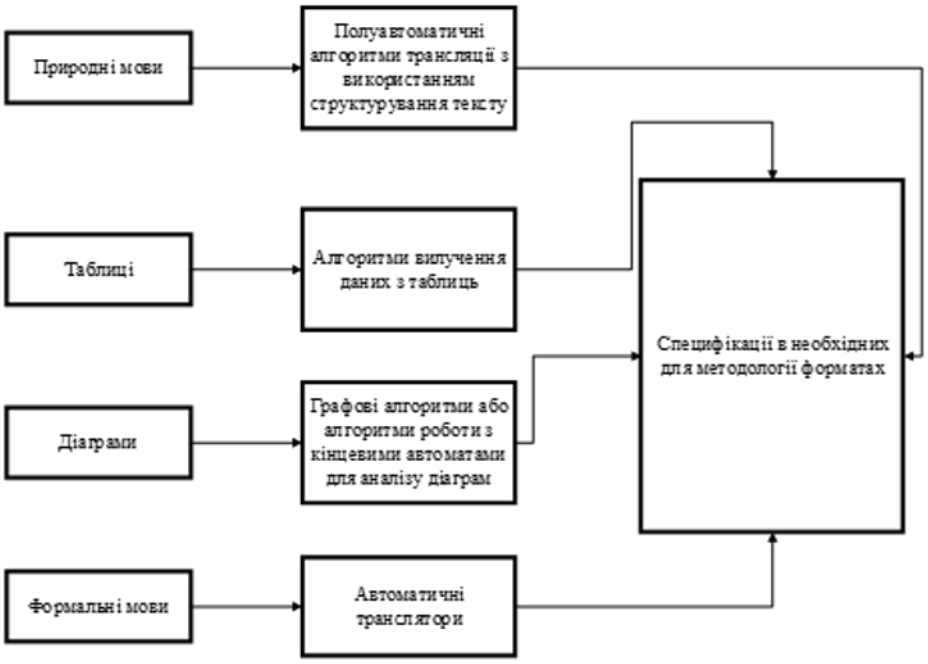
Аналіз форматів

Аналіз форматів

Більшість розглянутих форматів широко використовуються при створенні специфікацій обчислювальних систем, за винятком формальних мов. Однак, існують підходи до створення специфікацій і з використанням формальних мов.	Складність створення специфікацій прямо пропорційна ступеню формалізації використовуваної мови. З цієї причини необхідно шукати компроміс між якістю специфікації, безпосередньо залежать від її формалізації, і складністю її створення.	Можливості верифікації відсутні тільки у специфікацій, написаних на природних мовах, що і є однією з проблем, що вирішуються в даній роботі. Всі інші формати даних підтримують можливість перевірки коректності даних. У разі таблиць, однак, повноцінна верифікація вимагає створення таблиць по заздалегідь заданим шаблоном.	Можливості трансляції на формальні мови також відсутні тільки у специфікацій на природних мовах. У разі таблиць для такої трансляції також необхідно заздалегідь знати структуру таблиці. Для опису структурних аспектів архітектури СнК підходять природні мови, формальні мови та в меншій мірі структурні діаграми.

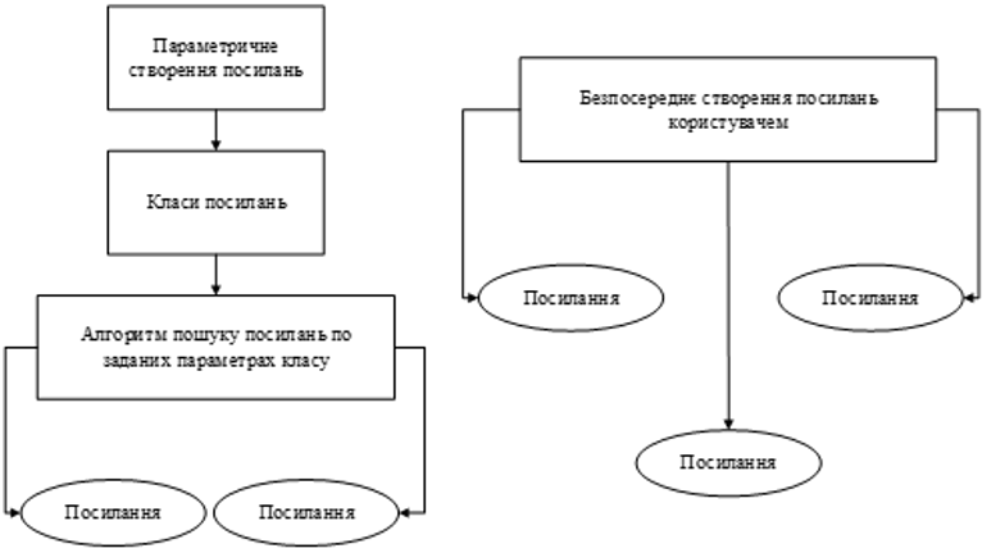
8

Трансляція даних



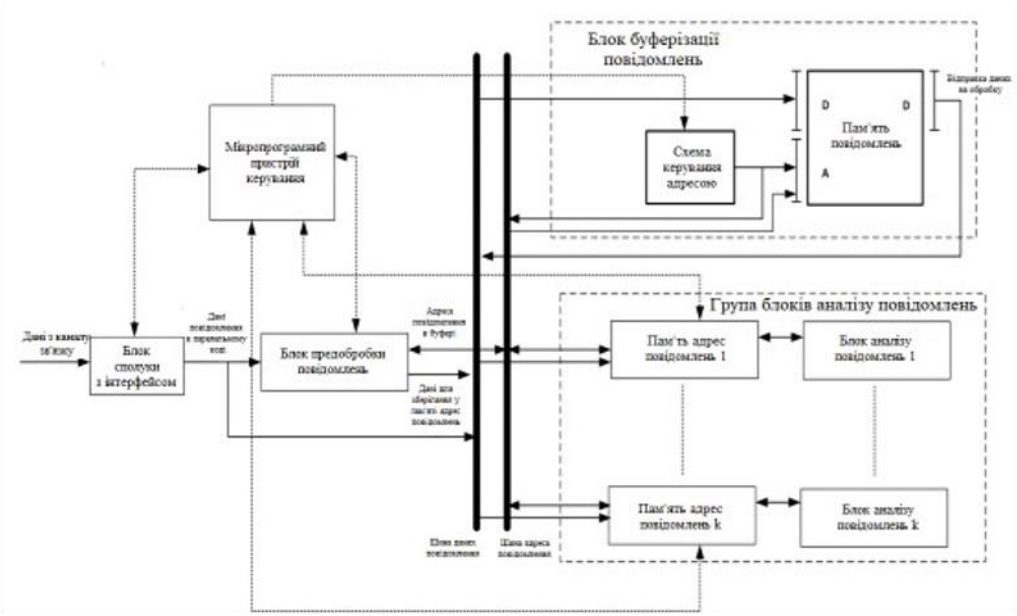
9

Процес створення посилань між фрагментами специфікацій



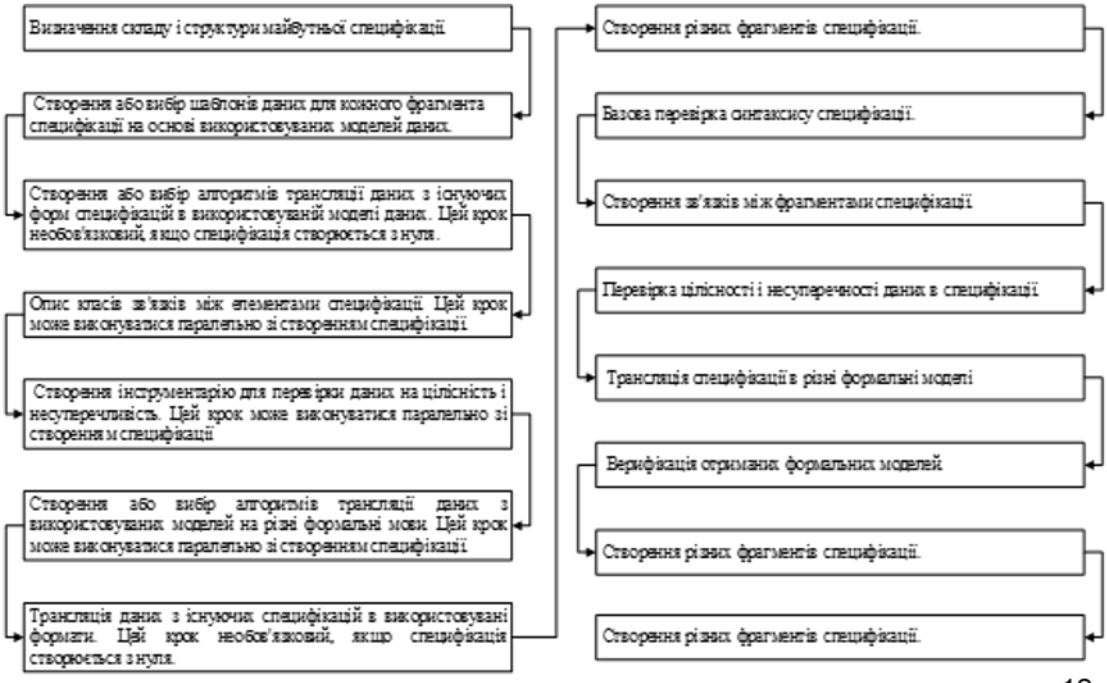
10

Структурна схема спеціалізованого пристрою управління обробки даних на основі ідентифікаторів



11

Метод створення специфікацій та роботи з ними



12

Результати роботи

SELECT bf.*, reg.ResetType FROM bitFields AS bf JOIN registers AS reg ON reg.RegisterType == bf.RegType WHERE bf.FieldName != "-"

Register Field Definitions						
Comments						Reset
FieldName	RegType	Bit Offset	Bit Width	Description	Access	ResetVal
-	PSTATE	13	51	Reserved	RO	0
tct	PSTATE	12	1	Trap on control transfer	RW	0
-	PSTATE	10	2	Reserved	RO	0
cle	PSTATE	9	1	Current little endian	RW	0
tle	PSTATE	8	1	Trap little endian	RW	0
-	PSTATE	6	2	Reserved	RO	0
-	PSTATE	5	1	Reserved (was red)	RO	0
pef	PSTATE	4	1	Enable floating-point	RW	1
am	PSTATE	3	1	Address mask	RW	0
priv	PSTATE	2	1	Privileged mode	RW	1
ie	PSTATE	1	1	Interrupt enable	RW	0
-	PSTATE	0	1	Reserved (was ag)	RO	0

Register Field Definitions							
Comments							Reset
FieldName	RegType	BitOffset	BitWidth	Description	Access	ResetType	ResetVal
gl	TSTATE	40	2	Global level at previous trap level	RW	PowerUp	0
ccr	TSTATE	32	8	CCR at previous trap level	RW	PowerUp	0
asi	TSTATE	24	8	ASI at previous trap level	RW	PowerUp	0
pstate.tct	TSTATE	20	1	PSTATE.tct at previous trap level	RW	PowerUp	0
pstate.cle	TSTATE	17	1	PSTATE.cle at previous trap level	RW	PowerUp	0
pstate.tle	TSTATE	16	1	PSTATE.tle at previous trap level	RW	PowerUp	0
pstate.pef	TSTATE	12	1	PSTATE.pef at previous trap level	RW	PowerUp	0
pstate.am	TSTATE	11	1	PSTATE.am at previous trap level	RW	PowerUp	0
pstate.priv	TSTATE	10	1	PSTATE.priv at previous trap level	RW	PowerUp	0
pstate.ie	TSTATE	9	1	PSTATE.ie at previous trap level	RW	PowerUp	0
cwp	TSTATE	0	3	CWP from previous trap level	RW	PowerUp	0
tct	PSTATE	12	1	Trap on control transfer	RW	PowerUp	0
cle	PSTATE	9	1	Current little endian	RW	PowerUp	0
tle	PSTATE	8	1	Trap little endian	RW	PowerUp	0
pef	PSTATE	4	1	Enable floating-point	RW	PowerUp	1
am	PSTATE	3	1	Address mask	RW	PowerUp	0
priv	PSTATE	2	1	Privileged mode	RW	PowerUp	1
ie	PSTATE	1	1	Interrupt enable	RW	PowerUp	0

```
reg {
name = "Processor State";
regwidth = 64;
reg {
name = "Processor State";
regwidth = 64;
reg {
name = "Processor State";
regwidth = 64;
field {
hw = rw; sw = rw;
fieldwidth = 1;
resetsignal = PowerUp;
reset = 0;
desc = Address mask;
} am [4:3]
```

13

Висновки

У ході підготовки кваліфікаційної роботи проведено аналіз методів створення архітектурних специфікацій обчислювальних комплексів на базі SoC. Для спрощення створення виконуваних моделей SoC у набір методів включений метод автоматизації переведення специфікацій у виконувані моделі на формальних мовах; а для раннього спільного аналізу різних компонентів системи та протоколів її функціонування розроблений підхід передбачає застосування різних методів верифікації даних на стадії створення специфікації . Крім того, розроблений метод також дозволяє підвищити якість створюваних специфікацій без суттєвих додаткових зусиль архітекторів та автоматизуватиперевикористання специфікацій обчислювальних систем.

14