AES
Embedded Systems Architecture

Registersatz; Implementierung von Speichern VHDL/FPGA

Harwarepraktikum

Louis Burda – Dirk Blödorn – Omer Faruk Aydin

5. Mai 2021

Nr.	Thema des Blattes	Gruppe
0	Wiederholung VHDL	
1	Instruktionsadressregister; Datenreplikationseinheit	
2	Registersatz; Implementierung von Speichern VHDL/FPGA	Do.1
3	Speicherinterface; Busanbindung	
4	Serielle Schnittstelle; Zustandsautomaten	
5	Takterzeugung; Systemtest	
6	Testbenches; Timing-Constraints	
7	Schiebeeinheit; generische iterative Beschreibungen	
8	Datenpfad; arithmetische & logische Operationen	
9	Execute-Stufe; Bypassing (Forwarding)	
10	Test des Gesamtsystems; Code-Generierung	

Tabelle 1: Theoriethemen

1 Einleitung

ARMv4 Prozessoren unterstützen verschiedene Betriebsmodi. Je nach aktuellem Modus hat der Prozessor die Sicht auf unterschiedliche Register. Im Folgenden untersuchen wir die sichtbaren Register abhängig vom Betriebsmodus und die Implementierung von solchen Speicherarten im FPGA.

2 ARMv4-ISA Betriebsmodi

CPU-Modus	Kürzel	THIS_MODE[4:0]	Beschreibung
User	usr	10000	regulärer, unpriviligierter Modus
FIQ	fiq	10001	schneller, priorisierter IRQ Handler
IRQ	irq	10010	regulärer IRQ Handler
Supervisor	svc	10011	behandelt Systemaufrufe/-initialisierung
Abort	abt	10111	behandelt Speicherzugriffsfehler
Undefined	und	11011	behandelt unbekannte Instruktionen
System	sys	11111	regulärer, priviligierter Modus

Tabelle 2: Betriebsmodi [2, A2-3]

Der Prozessor verbringt den Großteil der Zeit mit dem Ausführen von Instruktionen im unpriviligierte User Modus, welcher dadurch gekennzeichnet ist, dass nur durch das Auftreten eines Interrupts in einen priviligierten Modus gewechselt werden kann und dass Zugriffe auf bestimmte Speicherbereiche oder Prozessorfunktionen eventuell nicht erlaubt sind. Der System-Modus und die Ausnahmemodi (FIQ, IRQ, Supervisor, Abort, Undefined) sind priviligiert und können frei in einen anderen Modus wechseln. [2, A1-3]

3 ARMv4-ISA Registersatz

Die ARM Architektur enthält 31 general purpose register, wovon jedoch nur 16 Register sichtbar sind, zudem gibt es 6 Spezialregister.

In den Ausnahmemodi wird transparent ein Teil der Register mit dem Modus-eigenen Registern überlagert. Alle Ausnahmemodi besitzen eine eigene Abbildung für SP und LR, der Modus FIQ besitzt sogar eigene Versionen von R8 bis R12. Die erzeugte, modusabhängige Sicht auf Register vereinfacht den Moduswechsel und beschleunigt die Ausnahmebehandlung, da weniger Register gesichert werden müssen und somit langsame Speicherzugriffe vermieden werden. Wegen der hohen Anzahl an eigenen Registern und der hohen Priorisierung gilt deswegen auch der Modus FIQ im Gegensatz zum IRQ Modus als *Fast Interrupt Request*. [2, A2-6]

Die Register R0 - R12 sind beliebig nutzbar, sog. *general-purpose* Register. Im Register R13 wird der private Stackpointer des jeweiligen Betriebsmodus gespeichert, in R14 die

Rücksprungadresse zur Instruktion, zu welcher der Wechsel in den Betriebsmodus begann. Der Instruktionszähler (PC) wird im letzten Register R15 gespeichert. Der derzeitige Prozessorzustand liegt im CPSR Register, wobei der alte Zustand des Prozessors beim Wechsel des Prozessormodus im spezifischen SPSR gespeichert wird. [2, A2-6]

Das Setzen des USER-Bits in den Instruktionen LDM und STM hebt die Abbildung von der Modus-Spezifischen Register auf. Ein Zugriff auf Register R0 bis R15 bilden nun die Inhalte der Register des User Modus ab. Mit diesen Befehlen können mehrere Register aus dem User Modus gleichzeitig gesichert oder wiederhergestellt werden. [3]

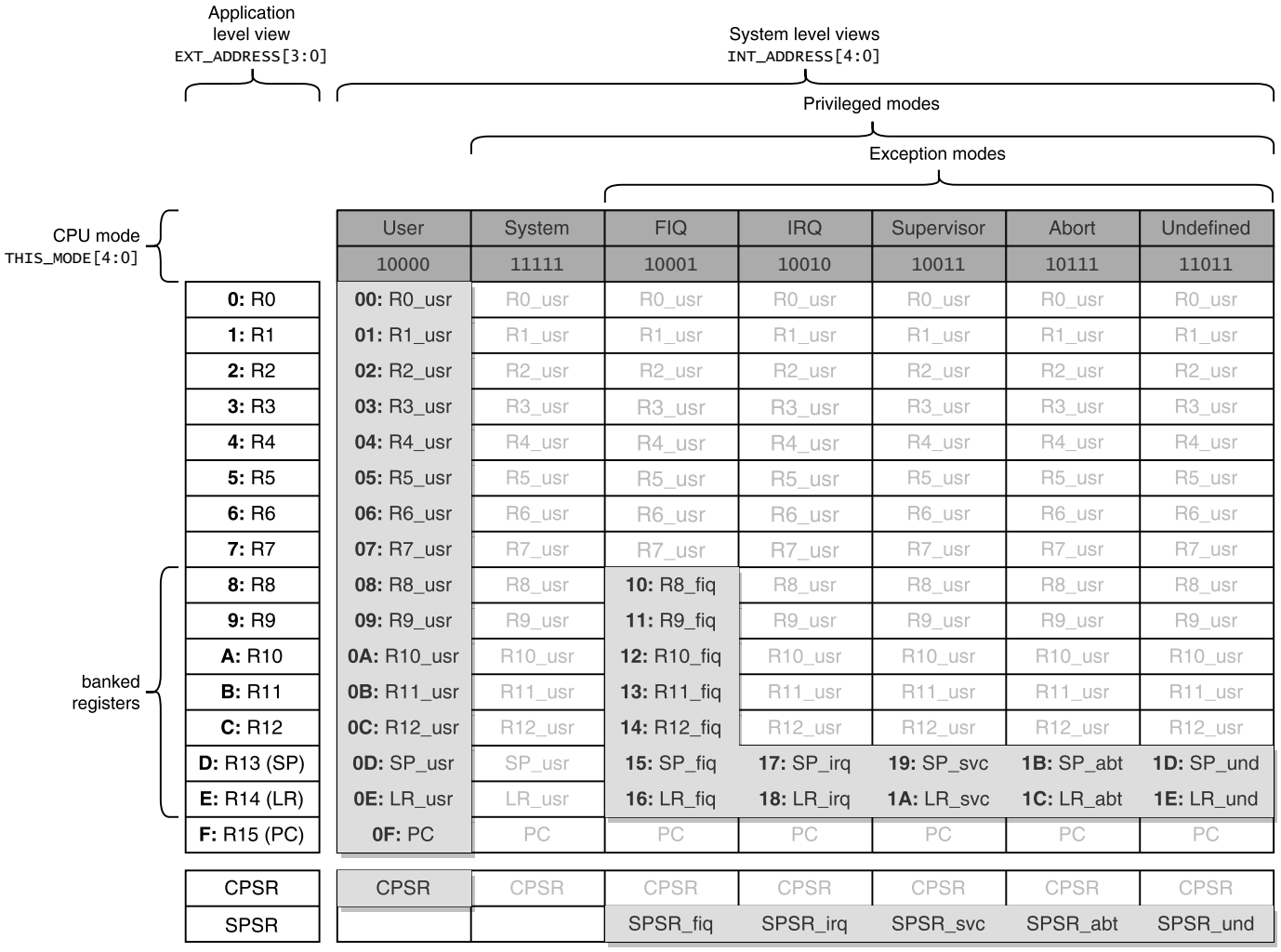

Abbildung 1: Übersetzung der internen auf externe Registeraddressen, in Abhängigkeit vom aktuellen CPU-Modus

4 Speicher in VHDL

Register können in VHDL mittels `std_logic_vector` implementiert werden, Memory kann direkt instanziiert werden oder als `array of std_logic_vector`. Memory ist zweidimensional angeordnet und zeichnet sich durch eine Adressierung eines Ausschnitts aus. Register bestehen aus getakteten FlipFlops und liegen physikalisch meist nah an der Logik, welche sie verwendet. Register sind sehr schnell, die Speicherdichte jedoch im Vergleich zu den meisten Memory-Instanziierungen gering. Im FPGA liegen die meisten Register in CLBs (Configurable Logic Blocks).

Memory ermöglicht je nach Implementierung eine hohe Speicherdichte. Operationen werden auf den Speicher durch ein Speicherinterface, für welches zusätzliche Logik nötig ist, durchgeführt. In Xilinx FPGAs der 7-Series kann er verteilt in LUTs (Distributed RAM) oder als Block RAM synthetisiert werden.

Das Basys 3 Artix-7 FPGA Board verwendet den Xilinx XC7A35T. [1]

	Distributed RAM	**Block RAM**
max. Größe[1]	400 Kb	1800 Kb
verteilt auf[1]	2400 Blöcke x 1b Weite	50 x 2b Weite
	x 64b Tiefe	x 36Kb Tiefe
Integrationsdichte	vergleichsweise gering	maximal
Implementierung	LUT (in SLICEM)[2]	als Hard-IP
Geschwindigkeit	hoch bei kleinen Speichern	von Entfernung
		zum CLB abhängig
Energieverbrauch	ultra low-power-modus möglich	gering
ECC-Fehlerprüfung	synthetisierbar	konfigurierbar

[1] bezogen auf den FPGA Xilinx XC7A35T im Basys 3 Artix-7 FPGA Board

[2] Die Slices unterscheiden sich zwischen SLICEL (L = logic), welche nur kombinatorische Logik unterstützen und SLICEM (M = memory), welche auch konfiguriert werden können, um distributed Memory oder Schieberegister abzubilden. Etwa 46% der Slices im XC7A35T sind SLICEM.

Tabelle 3: Unterschiede verschiedener Speicherarten

Literatur

[1] *7 Series FPGAs Data Sheet: Overview.* URL: `https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf`.

[2] *Arm Architecture Reference Guide.* Juli 2005.

[3] *ARM Befehlssatzarchitektur.* URL: `https://isis.tu-berlin.de/pluginfile.php/1915265/mod_resource/content/3/HWPR_ARM_Befehlssatzarchitektur_edit2.pdf`.